



Leitfaden

Amazon Simple Storage Service



API-Version 2006-03-01

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Simple Storage Service: Leitfaden

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon S3?	1
Funktionen von Amazon S3	1
Speicherklassen	1
Speicherverwaltung	2
Zugriffsverwaltung und Sicherheit	3
Datenverarbeitung	4
Speicherprotokollierung und Überwachung	5
Analytik und Einblicke	5
Starke Konsistenz	6
Funktionsweise von Amazon S3	6
Buckets	7
Objekte	8
Schlüssel	8
S3-Versioning	8
Versions-ID	9
Bucket-Richtlinie	9
S3-Zugriffspunkte	10
Zugriffskontrolllisten (ACLs)	10
Regionen	11
Amazon S3-Datenkonsistenzmodell	11
Gleichzeitige Anwendungen	12
Zugehörige Services	14
Zugriff auf Amazon S3	15
AWS Management Console	15
AWS Command Line Interface	15
AWS SDKs	15
Amazon S3 REST API	16
Bezahlen für Amazon S3	16
PCIDSSEinhaltung der Vorschriften	17
Erste Schritte	18
Einrichtung	19
Melde dich an für ein AWS-Konto	19
Erstellen eines Benutzers mit Administratorzugriff	20
Schritt 1: Erstellen eines Buckets	21

Schritt 2: Hochladen eines Objekts	28
Schritt 3: Herunterladen eines Objekts	29
Verwenden der S3-Konsole	29
Schritt 4: Kopieren eines Objekts	30
Schritt 5: Löschen von Objekt und Bucket	32
Löschen eines Objekts	33
Leeren des Buckets	33
Löschen des Buckets	34
Nächste Schritte	34
Verstehen allgemeiner Anwendungsfälle	35
Kontrollieren des Zugriffs auf Buckets und Objekte	35
Verwalten und Überwachen Ihres Speichers	36
Entwickeln mit Amazon S3	37
Erwerben Sie neue Kenntnisse aus Tutorials	38
Lernen Sie Trainings und Support kennen	40
Tutorials	41
Erste Schritte	39
Optimierung der Speicherkosten	39
Verwalten des Speichers	39
Hosten von Videos und Websites	39
Verarbeiten von Daten	39
Schutz von Daten	40
Transformieren von Daten mit S3 Object Lambda	42
Voraussetzungen	44
Schritt 1: Einen S3-Bucket erstellen	46
Schritt 2: Hochladen einer Datei zu einem S3-Bucket	47
Schritt 3: Erstellen eines S3-Zugriffspunkts	48
Schritt 4: Erstellen einer Lambda-Funktion	49
Schritt 5: Konfigurieren Sie eine IAM Richtlinie für die Ausführungsrolle Ihrer Lambda- Funktion	56
Schritt 6: Erstellen eines S3 Object Lambda Access Point	57
Schritt 7: Anzeigen der transformierten Daten	58
Schritt 8: Bereinigen	61
Nächste Schritte	64
Daten erkennen und redigieren PII	65
Voraussetzungen: Erstellen Sie einen IAM Benutzer mit Berechtigungen	67

Schritt 1: Einen S3-Bucket erstellen	69
Schritt 2: Hochladen einer Datei zu einem S3-Bucket	70
Schritt 3: Erstellen eines S3-Zugriffspunkts	71
Schritt 4: Konfigurieren und Bereitstellen einer vordefinierten Lambda-Funktion	72
Schritt 5: Erstellen eines S3 Object Lambda Access Point	73
Schritt 6: Verwenden des S3 Object Lambda Access Point zum Abrufen der redigierten Daten	75
Schritt 7: Bereinigen	76
Nächste Schritte	80
Hosting von Video-Streaming	81
Voraussetzungen: Registrieren und Konfigurieren einer benutzerdefinierten Domäne mit Route 53	83
Schritt 1: Einen S3-Bucket erstellen	84
Schritt 2: Hochladen eines Videos in den S3-Bucket	85
Schritt 3: Erstellen Sie eine ursprüngliche CloudFront Zugriffsidentität	86
Schritt 4: Erstellen Sie eine CloudFront Distribution	86
Schritt 5: Greifen Sie über die CloudFront Distribution auf das Video zu	89
Schritt 6: Konfigurieren Sie Ihre CloudFront Distribution so, dass sie Ihren benutzerdefinierten Domainnamen verwendet	90
Schritt 7: Greifen Sie über die CloudFront Distribution mit dem benutzerdefinierten Domainnamen auf das S3-Video zu	95
(Optional) Schritt 8: Zeigen Sie Daten zu Anfragen an, die bei Ihrer CloudFront Distribution eingegangen sind	96
Schritt 9: Bereinigen	96
Nächste Schritte	102
Batch-Transkodierung-Videos	102
Voraussetzungen	104
Schritt 1: Erstellen Sie einen S3-Bucket für Ausgabe-Mediendateien	105
Schritt 2: Erstelle eine IAM Rolle für MediaConvert	107
Schritt 3: Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion	108
Schritt 4: Erstellen einer Lambda-Funktion für die Videotranscodierung	110
Schritt 5: Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket	128
Schritt 6: Eine IAM Rolle für S3 Batch Operations erstellen	132
Schritt 7: Einrichten und Ausführen eines Auftrags für S3-Batchvorgänge	136
Schritt 8: Überprüfen Sie die Ausgabe-Mediendateien aus Ihrem S3-Ziel-Bucket	141
Schritt 9: Bereinigen	142

Nächste Schritte	145
Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität	145
Voraussetzungen	146
Schritt 1: Erstellen Sie eine große Datei	147
Schritt 2: Teilen Sie die Datei in mehrere Dateien auf	147
Schritt 3: Erstellen Sie den mehrteiligen Upload mit einer zusätzlichen Prüfsumme	149
Schritt 4: Laden Sie die Teile Ihres mehrteiligen Uploads hoch	150
Schritt 5: Listet alle Teile eures mehrteiligen Uploads auf	151
Schritt 6: Schließen Sie den mehrteiligen Upload ab	153
Schritt 7: Vergewissern Sie sich, dass das Objekt in Ihren Bucket hochgeladen wurde	154
Schritt 8: Überprüfen Sie die Objektintegrität mit einer MD5 Prüfsumme	154
Schritt 9: Überprüfen Sie die Objektintegrität mit einer zusätzlichen Prüfsumme	156
Schritt 10: Bereinigen Ihrer Ressourcen	158
Konfigurieren einer statischen Website	159
Schritt 1: Erstellen eines Buckets	160
Schritt 2: Aktivieren des statischen Website-Hostings	160
Schritt 3: Bearbeiten der Block-Public-Access-Einstellungen	162
Schritt 4: Hinzufügen einer Bucket-Richtlinie, die den Inhalt Ihres Buckets öffentlich verfügbar macht	163
Schritt 5: Konfigurieren eines Indextdokuments	165
Schritt 6: Konfigurieren eines Fehlerdokuments	166
Schritt 7: Testen des Website-Endpunkts	167
Schritt 8: Bereinigen	168
Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten Domäne	168
Bevor Sie beginnen	170
Schritt 1: Registrieren einer benutzerdefinierten Domäne bei Route 53	170
Schritt 2: Erstellen von zwei Buckets	171
Schritt 3: Konfigurieren des Stammdomänen-Buckets	172
Schritt 4: Konfigurieren des Subdomänen-Buckets für die Umleitung	174
Schritt 5: Konfigurieren der Protokollierung	174
Schritt 6: Hochladen des Index und des Website-Inhalts	175
Schritt 7: Hochladen eines Fehlerdokuments	177
Schritt 8: Bearbeiten der Block Public Access-Einstellungen	178
Schritt 9: Anfügen einer Bucket-Richtlinie	179
Schritt 10: Testen Ihres Domänen-Endpunkts	181

Schritt 11: Hinzufügen von Aliasdatensätzen	182
Schritt 12: Testen der Website	187
Beschleunigen Sie Ihre Website mit Amazon CloudFront	188
Bereinigen von Beispielressourcen	193
Arbeiten mit Buckets	196
Bucket-Übersicht	197
Berechtigungen	198
Verwalten des öffentlichen Zugriffs auf Buckets	199
Bucket-Konfiguration	201
Benennungsregeln	204
Regeln für die Benennung von Buckets für allgemeine Zwecke	204
Regeln für die Benennung von Verzeichnis-Buckets	207
Zugreifen auf einen Bucket und Auflisten des Buckets	208
.....	208
Auflisten eines Buckets	210
Erstellen eines Buckets	211
Anzeigen von Bucket-Eigenschaften	224
Leeren eines Buckets	228
Leeren eines Buckets mit AWS CloudTrail konfiguriert	230
Löschen eines Buckets	231
Festlegen der Standard-Bucket-Verschlüsselung	236
Verwendung von SSE — KMS Verschlüsselung für kontoübergreifende Operationen	238
Verwenden der Standard-Verschlüsselung mit der Replikation	239
Verwenden von Amazon S3-Bucket-Schlüsseln mit Standard-Verschlüsselung	240
Konfigurieren der Standardverschlüsselung	240
Überwachen der Standard-Verschlüsselung	246
Mountpoint für Amazon S3	248
Mountpoint installieren	248
Mountpoint konfigurieren und verwenden	254
Konfigurieren der Transferbeschleunigung	257
Gründe für die Nutzung von Amazon S3 Transfer Acceleration	258
Anforderungen für die Verwendung von Transfer Acceleration	258
Erste Schritte	260
Aktivieren der Transfer Acceleration	262
Speed Comparison Tool	269
Verwenden von Zahlung durch den Anforderer	270

Die Gebühren bei Zahlung durch den Anforderer	272
Konfigurieren von Zahlung durch den Anforderer	272
Die Konfiguration wird abgerufen requestPayment	274
Objekte aus Requester Pays-Buckets werden heruntergeladen	275
Kontingente, Einschränkungen und Beschränkungen	276
Arbeiten mit Objekten	279
Objekte	280
Subressourcen	282
Erstellen von Objektschlüsseln	282
Richtlinien für Objektschlüsselnamen	283
Arbeiten mit Metadaten	287
Systemdefinierte Objektmetadaten	287
Benutzerdefinierte Objektmetadaten	291
Objektmetadaten bearbeiten	293
Objekte hochladen	296
Hochladen eines Objekts	298
Verhindern Sie das Hochladen von Objekten mit identischen Schlüsselnamen	311
Verwenden von mehrteiligen Uploads	311
Mehrteiliger Upload-Prozess	312
Prüfsummen mit mehrteiligen Upload-Operationen	315
Gleichzeitige mehrteilige Upload-Vorgänge	315
Verhindern Sie, dass beim mehrteiligen Upload Objekte mit identischen Schlüsselnamen hochgeladen werden	316
Mehrteiliger Upload und Preise	316
API-Unterstützung für mehrteilige Uploads	317
AWS Command Line Interface Unterstützung für mehrteiligen Upload	318
AWS SDK-Unterstützung für mehrteilige Uploads	318
Mehrteiliger Upload API und Berechtigungen	319
Konfigurieren einer Lebenszykluskonfiguration	323
Hochladen eines Objekts mit Multipart-Upload	327
Hochladen eines Verzeichnisses	351
Auflisten von mehrteiligen Uploads	354
Verfolgen eines mehrteiligen Uploads	356
Abbrechen eines mehrteiligen Uploads	360
Kopieren eines Objekts	366
Limits mehrteiliger Uploads	372

Bedingte Anfrage stellen	373
Wie ruft man Objekte auf der Grundlage von Metadaten ab oder kopiert sie	374
Wie verhindert man Objektüberschreibungen	376
Objekte kopieren, verschieben und umbenennen	380
Ein Objekt kopieren	383
So verschieben Sie ein Objekt	394
Um ein Objekt umzubenennen	396
Herunterladen von Objekten	397
Herunterladen eines Objekts	398
Herunterladen mehrerer Objekte	400
Herunterladen eines Teils eines Objekts	402
Ein Objekt von einem anderen herunterladen AWS-Konto	403
Herunterladen von archivierten Objekten	404
Objekte auf der Grundlage von Metadaten herunterladen	404
Fehlerbehebung beim Herunterladen von Objekten	405
Überprüfung der Objektintegrität	405
Verwenden unterstützter Prüfsummenalgorithmen	405
Inhalte verwenden — MD5 beim Hochladen von Objekten	415
Verwenden Sie Content-MD5 und the, um hochgeladene ETag Objekte zu verifizieren	415
Verwenden von nachfolgenden Prüfsummen	416
Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads	416
Löschen von Objekten	418
Programmgesteuertes Löschen von Objekten aus einem Bucket mit aktivierter Versionierung	419
Löschen von Objekten aus einem Bucket mit aktivierter Option MFA	419
Löschen eines einzelnen Objekts	420
Löschen von mehreren Objekten	432
Organisieren und Auflisten von Objekten	435
Verwenden von Präfixen	436
Auflisten von Objekten	438
Verwenden von Ordnern	441
Anzeigen einer Objektübersicht	445
Anzeigen von Objekteigenschaften	446
Arbeitet mit presigned URLs	448
Wer kann ein vorsigniertes erstellen URL	449
Ablaufzeit für vorsignierte URLs	450

Einschränkung vordefinierter Funktionen URL	451
Objekte mit vorsignierten teilen URLs	452
Objekte mit Presigned hochladen URLs	456
Transformieren von Objekten	458
Erstellen von Objekt-Lambda-Zugriffspunkten	460
Verwenden von Amazon S3 Objekt-Lambda-Zugriffspunkten	476
Sicherheitsüberlegungen	480
Schreiben von Lambda-Funktionen	487
Die Verwendung von AWS gebaute Funktionen	520
Bewährte Methoden und Richtlinien für S3 Object Lambda	522
Tutorials zu S3 Object Lambda	524
Debuggen von S3 Object Lambda	524
Was ist S3 Express One Zone?	526
Übersicht	528
Einzelne Availability Zone	528
Verzeichnis-Buckets	528
Endpunkte und Gateway-Endpunkte VPC	529
Sitzungsbasierte Autorisierung	529
Funktionen von S3 Express One Zone	530
Zugriffsverwaltung und Sicherheit	530
Protokollierung und Überwachung	531
Verwaltung von Objekten	532
AWS SDKsund Client-Bibliotheken	532
Verschlüsselung und Datenschutz	533
AWS Signatur Version 4 (SigV4)	533
Starke Konsistenz	533
Zugehörige Services	533
Nächste Schritte	535
Wodurch zeichnet sich S3 Express One Zone aus?	535
Unterschiede von S3 Express One Zone	536
API von S3 Express One Zone unterstützte Operationen	539
Amazon-S3-Features, die von S3 Express One Zone nicht unterstützt werden	540
Tutorial: Erste Schritte mit S3 Express One Zone	542
Voraussetzungen	543
Schritt 1: Konfigurieren Sie einen VPC Gateway-Endpunkt	546
Schritt 2: Erstellen Sie einen Verzeichnis-Bucket	548

Schritt 3: Daten in einen Verzeichnis-Bucket importieren	551
Schritt 4: Laden Sie Objekte manuell in Ihren Verzeichnis-Bucket hoch	553
Schritt 5: Leeren Sie Ihren Verzeichnis-Bucket	554
Schritt 6: Löschen Sie Ihren Verzeichnis-Bucket	555
Nächste Schritte	555
Networking für S3 Express One Zone	557
Endpunkte	557
Konfiguration von VPC-Gateway-Endpunkten	558
Verzeichnis-Buckets	559
Availability Zones	560
Namen von Verzeichnis-Buckets	561
Verzeichnisse	561
Schlüsselnamen	562
Zugriffsverwaltung	562
Arbeiten mit Verzeichnis-Buckets	562
Regeln für die Benennung von Verzeichnis-Buckets	563
Erstellen eines Verzeichnis-Buckets	564
Anzeigen von Eigenschaften	574
Verwalten von Bucket-Richtlinien	575
Leeren eines Verzeichnis-Buckets	580
Löschen eines Verzeichnis-Buckets	582
Auflisten von Verzeichnis-Buckets	584
Beispiele für HeadBucket	587
Arbeiten mit Objekten in einem Verzeichnis-Bucket	588
Importieren von Objekten in einen Verzeichnis-Bucket	589
Verwenden von Batch Operations mit S3 Express One Zone	591
Hochladen eines Objekts	594
Mehrteilige Uploads mit Verzeichnis-Buckets verwenden	597
Kopieren eines Objekts	627
Löschen eines Objekts	633
Herunterladen eines Objekts	637
Beispiele für HeadObject	639
Sicherheit für Verzeichnis-Buckets	640
Datenschutz und Verschlüsselung in S3 Express One Zone	641
IAM für S3 Express One Zone	667
Identitätsbasierte Richtlinien	686

Bucket-Richtlinien	687
CreateSession-Autorisierung	689
Bewährte Methoden für die Sicherheit	691
Protokollierung mit AWS CloudTrail für S3 Express One Zone	695
CloudTrail Verwaltungsereignisse für S3 Express One Zone	695
CloudTrail Datenereignisse für S3 Express One Zone	696
.....	698
Optimieren der Leistung von S3 Express One Zone	702
Leistungsrichtlinien und Entwurfsmuster	703
Entwicklung mit S3 Express One Zone	707
Availability Zones und Regionen bei S3 Express One Zone	708
Regionale und zonale Endpunkte	710
Arbeiten mit S3 Express One Zone mithilfe der S3-Konsole AWS CLI, und AWS SDKs	710
S3 Express API One-Zone-Betrieb	712
Arbeiten mit Zugriffspunkten	715
IAMRichtlinien konfigurieren	716
Beispiele von -Zugriffspunktrichtlinien	717
Bedingungsschlüssel	721
Delegieren der Zugangskontrolle an Zugriffspunkte	722
Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte	723
Erstellen von Zugriffspunkten	724
Regeln zur Benennung von Amazon S3-Zugriffspunkten	724
Erstellen eines Zugriffspunkts	725
Erstellen von Zugangspunkten, die auf a beschränkt sind VPC	728
Verwalten des öffentlichen Zugriffs	730
Verwenden von Zugriffspunkten	732
Zugreifen auf einen Bucket über einen S3-Zugriffspunkt	733
Überwachung und Protokollierung	733
Verwalten von Zugriffspunkten	735
Verwenden eines Alias im Bucket-Stil für Ihren Zugriffspunkt.	738
Verwenden von Zugriffspunkten mit Amazon-S3-Operationen	741
Beschränkungen und Einschränkungen	744
Arbeiten mit Multi-Regions-Zugriffspunkten	747
Erstellen Multi-Regions-Zugriffspunkten	748
Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten	750
Regeln für die Auswahl von Buckets für Amazon S3-Multi-Regions-Zugriffspunkte	751

Erstellen eines Amazon S3-Multi-Region Access Point	753
Blockieren des öffentlichen Zugriffs mit Amazon S3-Multi-Regions-Zugriffspunkten	755
Anzeigen der Konfigurationsdetails für Amazon S3 Multi-Region Access Points	756
Löschen eines Multi-Region Access Point	758
Konfigurieren von Multi-Region Access Points	759
Konfiguration AWS PrivateLink	760
Entfernen des Zugriffs auf einen Multi-Region Access Point von einem VPC-Endpunkt	763
Verwenden von Multi-Region Access Points	763
Hostnamen für Multi-Regions-Zugriffspunkte	765
Multi-Regions-Zugriffspunkte und Amazon S3 Transfer Acceleration	766
Berechtigungen	767
Beschränkungen und Einschränkungen	775
Weiterleitung von Anforderungen	778
Failover-Konfiguration	780
Bucket-Replikation	788
Unterstützte Operationen API	798
Überwachung und Protokollierung	815
Sicherheit	819
Datenschutz	820
Datenverschlüsselung	822
Server-side encryption	824
Verwenden der clientseitigen Verschlüsselung	918
Datenschutz zwischen Netzwerken	919
Datenverkehr zwischen Service und lokalen Clients und Anwendungen	919
Verkehr zwischen AWS Ressourcen in derselben Region	920
AWS PrivateLink für Amazon S3	920
Typen von VPC-Endpunkten	921
Einschränkungen und Beschränkungen von AWS PrivateLink für Amazon S3	922
Einen VPC Endpunkt erstellen	922
Zugriff auf Amazon-S3-Schnittstellen-Endpunkte	923
Privat DNS	923
Zugriff auf Buckets, Access Points und Amazon S3 API Control-Operationen von Endpunkten mit S3-Schnittstelle	926
Aktualisierung einer lokalen Konfiguration DNS	933
Eine VPC Endpunktrichtlinie erstellen	935
Zugriffsverwaltung	938

S3-Ressourcen	939
Identitäten	945
Tools zur Zugriffsverwaltung	948
Aktionen	954
Anwendungsfälle für die Zugriffsverwaltung	955
Fehlerbehebung bei der Zugriffsverwaltung	963
Identitäts- und Zugriffsverwaltung	965
Verwalten des Zugriffs mit S3-Zugriffsberechtigungen	1187
Verwaltung des Zugriffs mit ACLs	1301
Blockieren des öffentlichen Zugriffs	1347
Überprüfen des Bucket-Zugriffs	1366
Überprüfen der Bucket-Eigentümerschaft	1374
Steuern der Objekteigentümerschaft	1379
Protokollierung und Überwachung	1425
Compliance-Validierung	1428
Ausfallsicherheit	1430
Verschlüsselung von Sicherungen	1432
Sicherheit der Infrastruktur	1433
Konfigurations- und Schwachstellenanalyse	1434
Bewährte Methoden für die Gewährleistung der Sicherheit	1435
Bewährte Methoden für die Sicherheit in Amazon S3	1435
Bewährte Methoden zur Überwachung und Prüfung von Amazon S3	1442
Überwachung der Datensicherheit	1447
Verwalten des Speichers	1452
Verwenden der S3-Versioning	1453
Nicht versionierte, versionings-fähige und Buckets mit ausgesetztem Versioning	1453
Verwenden des S3-Versioning mit dem S3-Lebenszyklus	1454
S3-Versioning	1455
Aktivieren des Versioning für Buckets	1459
MFALöschen konfigurieren	1467
Arbeiten mit versioning-fähigen Objekten	1470
Arbeiten mit Objekten mit ausgesetztem Versioning	1501
Verwenden von AWS Backup für Amazon S3	1505
Arbeiten mit archivierten Objekten	1506
Wiederherstellen von Objekten aus S3 Glacier	1507
Wiederherstellen von Objekten aus S3 Intelligent-Tiering	1508

Verwenden von S3-Batch-Operationen mit Wiederherstellungsanforderungen	1508
Wiederherstellungszeit	1508
Archiv-Abrufoptionen	1509
Wiederherstellen eines archivierten Objekts	1512
Verwenden der Objektsperre	1521
So funktioniert die S3-Objektsperre	1522
Überlegungen zu Object Lock	1527
Konfigurieren der Objektsperre	1532
Verwalten von Speicherklassen	1544
Häufig aufgerufene Objekte	1544
Automatische Optimierung von Daten mit sich ändernden oder unbekanntem	
Zugriffsmustern	1545
Selten aufgerufene Objekte	1547
Objekte, auf die selten zugegriffen wird	1549
Amazon S3 in Outposts	1550
Vergleich der Speicherklassen	1551
Einrichten der Speicherklasse eines Objekts	1552
Amazon S3 Glacier-Speicherklassen	1554
Vergleich der S3 Glacier-Speicherklassen	1554
S3 Glacier Instant Retrieval	1555
S3 Glacier Flexible Retrieval	1555
S3 Glacier Deep Archive	1556
Archivierter Speicher	1557
Worin unterscheiden sich diese Speicherklassen vom S3 Glacier-Service	1558
Amazon S3 Intelligent Tiering	1558
So funktioniert S3 Intelligent-Tiering	1559
Verwenden von S3 Intelligent-Tiering	1563
Verwenden von S3 Intelligent-Tiering	1568
Verwalten des Lebenszyklus	1572
Verwaltung des gesamten Lebenszyklus von Objekten	1573
Übergang von Objekten	1574
Auslaufende Objekte	1584
Einrichten der Lebenszyklus-Konfiguration	1587
Verwenden anderer Bucket-Konfigurationen	1606
Konfigurieren von Lebenszyklus-Ereignisbenachrichtigungen	1609
Elemente der Lebenszyklus-Konfiguration	1611

Beispiele der S3-Lebenszyklus-Konfiguration	1624
Verwalten des Bestands	1644
Amazon-S3-Inventory-Buckets	1645
Bestandslisten	1646
Konfigurieren von Amazon S3 Inventory	1652
Lokalisieren Ihres Bestands	1663
Einrichten von Benachrichtigungen für den Bestandsabschluss	1667
Bestandsabfrage mit Athena	1668
Konvertieren leerer Versions-ID-Zeichenfolgen in Null-Zeichenfolgen	1674
Arbeiten mit dem Feld „Objekt-ACL“	1677
Replizieren von Objekten	1679
Warum sollten Sie Replikation verwenden?	1681
Verwenden der regionsübergreifenden Replikation	1682
Verwenden von Replikation innerhalb derselben Region	1682
Wann sollte die bidirektionale Replikation verwendet werden	1683
Wann die S3-Batch-Replikation verwendet wird	1684
Workload-Anforderungen und Live-Replikation	1684
Was wird repliziert?	1685
Anforderungen und Überlegungen für die Replikation	1690
Live-Replikation einrichten	1693
Live-Replikation verwalten oder pausieren	1786
Replizieren vorhandener Objekte	1788
Den Fortschritt überwachen und den Status abrufen	1802
Verwenden von Objekt-Markierungen	1827
API-Operationen für die Objektmarkierung	1830
Zusätzliche Konfigurationen	1831
Zugriffskontrolle	1832
Verwalten von Objekt-Markierungen	1836
Verwenden von Kostenzuordnungs-Markierungen	1841
Weitere Infos	1843
Fakturierungs- und Nutzungsberichte	1843
Fakturierungsberichte	1844
Nutzungsbericht	1847
Fakturierungs- und Nutzungsberichte verstehen	1850
Abrechnung für Amazon S3 S3-Fehlerantworten	1878
Verwenden von Amazon S3 Select	1903

Voraussetzungen und Einschränkungen	1903
Erstellen einer Anforderung	1904
Fehler	1906
S3-Select-Beispiele	1906
SQL-Referenz	1910
Verwendung von Batchvorgänge	1952
Grundlagen von Batchvorgänge	1953
Tutorial zu S3-Batchvorgängen	1954
Gewähren von Berechtigungen	1954
Erstellen eines-Auftrags	1965
Unterstützte Vorgänge	1989
Verwalten von Aufträgen	2033
Verfolgen von Auftragsstatus- und Abschluss	2038
Verwenden von Markierungen	2055
Verwalten der S3-Objektsperre	2072
Tutorial zu S3-Batchvorgängen	2096
Überwachen von Amazon S3	2097
Überwachungstools	2098
Automatisierte Tools	2098
Manuelle Tools	2098
Protokollierungsoptionen	2099
Protokollierung mit CloudTrail	2102
Verwenden von CloudTrail Protokollen mit Amazon S3 S3-Serverzugriffsprotokollen und CloudWatch Protokollen	2104
CloudTrail Nachverfolgung mit Amazon S3 SOAP API S3-Anrufen	2105
CloudTrail Ereignisse	2106
Beispiele für Protokolldateien	2119
Aktivieren CloudTrail	2125
Identifizieren von S3-Anfragen	2129
Protokollierungs-Serverzugriff	2137
Wie aktiviere ich die Protokollzustellung?	2137
Protokollobjekt-Schlüsselformat	2140
Wie werden Protokolle ausgeliefert?	2141
Best-Effort-Serverprotokollbereitstellung	2142
Statusänderungen in der Bucket-Protokollierung werden mit der Zeit wirksam	2142
Aktivieren der Server-Zugriffsprotokollierung	2143

Protokollformat	2165
Löschen von Protokolldateien	2180
Identifizieren von S3-Anfragen	2181
Metriken überwachen mit CloudWatch	2187
Metriken und Dimensionen	2190
Zugriff auf CloudWatch Metriken	2209
CloudWatch Metriken, Konfigurationen	2210
Amazon-S3-Ereignis-Benachrichtigungen	2220
Übersicht	2220
Benachrichtigungstypen und -ziele	2222
Verwenden von SQSSNS, und Lambda	2231
Benutzen EventBridge	2262
Verwenden von Analysen und Einblicken	2273
Speicherklassen-Analyse	2273
Einrichten der Speicherklassen-Analyse	2274
Speicherklassen-Analyse	2275
Wie kann ich die Daten der Speicherklassen-Analyse exportieren?	2277
Konfigurieren der Speicherklassen-Analyse	2278
S3 Storage Lens	2281
Metriken und Funktionen von S3 Storage Lens	2282
Grundlegendes zu S3 Storage Lens	2284
Arbeiten mit Organisationen	2296
Berechtigungen für S3 Storage Lens	2300
Anzeigen von Speichermetriken	2304
Anwendungsfälle für Metriken von Amazon S3 Storage Lens	2338
Metrikglossar	2367
Arbeiten mit S3 Storage Lens	2404
Arbeiten mit S3-Storage-Lens-Gruppen	2454
Nachverfolgen von Anforderungen mit X-Ray	2496
So funktioniert X-Ray mit Amazon S3	2496
Verfügbare Regionen	2497
Hosten einer statischen Website	2498
Website-Endpunkte	2499
Website-Endpunkt-Beispiele	2500
Hinzufügen eines DNS CNAME	2501
Verwenden einer benutzerdefinierten Domäne mit Route 53	2501

Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt ...	2502
Aktivieren des Website-Hostings	2503
Konfigurieren eines Indextdokuments	2508
Indextdokument und Ordner	2509
Konfigurieren eines Indextdokuments	2510
Konfigurieren eines benutzerdefinierten Fehlerdokuments	2512
Amazon S3 HTTP S3-Antwortcodes	2512
Konfigurieren eines benutzerdefinierten Fehlerdokuments	2515
Festlegen von Berechtigungen für den Website-Zugriff	2516
Schritt 1: Bearbeiten der S3 Block Public Access-Einstellungen	2517
Schritt 2: Hinzufügen einer Bucket-Richtlinie	2519
Objektzugriffskontrolllisten	2521
Protokollieren des Webdatenverkehrs	2522
Konfigurieren einer Umleitung	2523
Anforderungen an einen anderen Host umleiten.	2524
Konfigurieren von Umleitungsregeln	2525
So leiten Sie Anforderungen für ein Objekt um	2533
Verwenden CORS	2535
Cross-Origin Resource Sharing: Szenarien in Anwendungsfällen	2535
Wie bewertet Amazon S3 die CORS Konfiguration eines Buckets?	2536
So unterstützt Object Lambda Access Point CORS	2537
Elemente einer CORS Konfiguration	2537
Konfiguration CORS	2543
Testen CORS	2552
Fehlerbehebung für CORS	2554
Entwickeln mit Amazon S3	2560
Senden von Anforderungen	2560
Über Zugriffsschlüssel	2561
Anforderungsendpunkte	2563
Senden von Anfragen über IPv6	2563
Senden von Anfragen unter Verwendung der AWS-SDKs	2574
Anfragen stellen mit dem REST API	2617
Verwendung der AWS CLI	2632
Verwendung der SDKs AWS	2634
Mit AWS SDKs arbeiten	2634
SDK-Programmierschnittstellen	2636

Angabe der Signature-Version in der Anforderungs-Authentifizierung	2636
Verwenden der REST-API	2647
Weiterleitung von Anforderungen	2648
Fehlerbehandlung	2655
Die REST-Fehlerantwort	2655
Die SOAP-Fehlerantwort	2657
Bewährte Methoden für Amazon-S3-Fehler	2658
Referenz	2659
Anhang A: Verwenden der SOAP-API	2660
Anhang b: Authentifizieren von Anfragen (AWS Signaturversion 2)	2665
Optimieren der Leistung von Amazon S3	2713
Leistungsanleitungen	2714
Messen der Leistung	2715
Horizontale Skalierung	2716
Verwenden von Byte Range Fetches	2716
Wiederholungsanforderungen	2716
Kombinieren Sie Amazon S3 und Amazon EC2 in derselben Region	2717
Verwenden der Transfer Acceleration zur Minimierung der Latenz	2717
Verwenden Sie die neueste AWS SDKs	2717
Leistungsdesignmuster	2718
Caching von Inhalten mit häufigen Zugriffen	2718
Timeouts und Wiederholungsversuche für latenzsensitive Anwendungen	2719
Horizontale Skalierung und Anforderungsparallelisierung	2720
Beschleunigung geographisch disparater Datenübertragungen	2722
Was ist S3 on Outposts?	2723
Funktionsweise von S3 on Outposts	2723
Regionen	2724
Buckets	2724
Objekte	2725
Schlüssel	2726
S3-Versioning	2726
Versions-ID	2726
Speicherklasse und Verschlüsselung	2727
Bucket-Richtlinie	2727
S3-on-Outposts-Zugriffspunkte	2728
Funktionen von S3 on Outposts	2728

Zugriffsverwaltung	2728
Speicherprotokollierung und Überwachung	2729
Starke Konsistenz	2729
Zugehörige Services	2730
Zugriff auf S3 on Outposts	2730
AWS Management Console	2730
AWS Command Line Interface	2731
AWS SDKs	2731
Bezahlung für S3 on Outposts	2731
Nächste Schritte	2732
Einrichten Ihres Outposts	2732
Einen neuen -Outpost bestellen	2732
Inwieweit S3 on Outposts anders ist	2733
Technische Daten	2733
Unterstützte API-Operationen	2734
Nicht unterstützte Amazon-S3-Funktionen	2734
Netzwerkeinschränkungen	2735
Erste Schritte mit S3 on Outposts	2736
Einrichten von IAM	2736
Verwenden der S3-Konsole	2745
Verwenden des AWS CLI Sand-SDK SDK for Java	2748
Vernetzung für S3 on Outposts	2753
Auswählen des Netzwerkzugriffstyps	2753
Zugriff auf Ihre S3-on-Outposts-Buckets und -Objekte	2754
Verwalten von Verbindungen mit kontenübergreifenden Elastic Network-Schnittstellen	2754
Arbeiten mit S3-on-Outposts-Buckets	2755
Buckets	2755
Zugriffspunkte	2755
Endpunkte	2756
API-Vorgänge in S3 on Outposts	2756
Erstellen und Verwalten von S3 on Outposts-Buckets	2758
Erstellen eines Buckets	2758
Hinzufügen von Tags	2763
Verwenden von Bucket-Richtlinien	2764
Auflisten von Buckets	2773
Abrufen eines Buckets	2775

Löschen des Buckets	2776
Arbeiten mit Zugriffspunkten	2778
Arbeiten mit Endpunkten	2792
Arbeiten mit S3-on-Outposts-Objekten	2798
Hochladen eines Objekts	2800
Kopieren eines Objekts	2802
Ein Objekt abrufen	2804
Auflisten von Objekten	2807
Löschen von Objekten	2810
Verwendung von HeadBucket	2815
Durchführen eines mehrteiligen Uploads	2817
Verwenden vorsegnierter URLs	2824
Amazon S3 on Outposts mit lokalem Amazon EMR	2838
Zwischenspeicherung von Autorisierung und Authentifizierung	2845
Sicherheit	2847
Datenverschlüsselung	2848
AWS PrivateLink für S3 auf Outposts	2848
Signature Version 4 (SigV4) Richtlinienschlüssel	2855
Von AWS verwaltete Richtlinien	2859
Verwenden von serviceverknüpften Rollen	2861
Verwaltung von S3-on-Outposts-Speicher	2866
Verwalten der S3-Versioning	2866
Erstellen und Verwalten einer Lebenszyklus-Konfiguration	2869
Replikation von Objekten für S3 in Outposts	2878
Freigabe von S3 on Outposts	2911
Sonstige Services	2916
Überwachen von S3 in Outposts	2917
CloudWatch Metriken	2917
CloudWatch Amazon-Veranstaltungen	2920
CloudTrail Protokolle	2921
Entwickeln mit S3 on Outposts	2924
APIs für S3 on Outposts	2925
Konfigurieren des S3-Steuerungs-Clients	2928
Senden von Anforderungen über IPv6	2928
Codebeispiele	2940
Grundlagen	2957

Hello Amazon S3	2960
Erlernen Sie die Grundlagen	2972
Aktionen	3062
Szenarien	3511
Text in Sprache und zurück in Text konvertieren	3513
Erstellen Sie ein vorsigniertes URL	3514
Erstellen einer Serverless-Anwendung zur Verwaltung von Fotos	3553
Eine Webseite erstellen, die Amazon-S3-Objekte auflistet	3558
Erstellen Sie eine Amazon-Textextract-Explorer-Anwendung	3560
Löscht alle Objekte in einem Bucket	3561
Löschen Sie unvollständige mehrteilige Uploads	3563
In Bildern erkennen PPE	3567
Entitäten in Text erkennen, der aus einem Bild extrahiert wurde	3568
Gesichter in einem Bild erkennen	3568
Erkennen von Objekten in Bildern	3569
Erkennen Sie Personen und Objekte in einem Video	3573
Herunterladen von Objekten in ein lokales Verzeichnis	3573
Rufen Sie ein Objekt von einem Access Point mit mehreren Regionen ab	3575
Abrufen eines Objekts aus einem Bucket abrufen, wenn es geändert wurde	3576
Erste Schritte mit der Verschlüsselung	3581
Erste Schritte mit Tags	3587
Amazon S3 S3-Objekte sperren	3590
Zugriffskontrolllisten verwalten () ACLs	3676
Versionierte Objekte in Batches mit einer Lambda-Funktion verwalten	3682
Analysieren URIs	3683
Erstellen einer mehrteiligen Kopie	3686
Durchführen eines mehrteiligen Uploads	3689
S3-Ereignisbenachrichtigungen verarbeiten	3693
Speichern EXIF und andere Bildinformationen	3696
Senden Sie Ereignisbenachrichtigungen an EventBridge	3697
Verfolgen Sie Uploads und Downloads	3700
Transformieren Sie Daten mit S3 Object Lambda	3703
Einheiten- und Integrationstest mit einem SDK	3704
Hochladen eines Verzeichnisses in einen Bucket	3712
Hoch- oder Herunterladen großer Dateien	3713
Stream unbekannter Größe hochladen	3754

Verwenden der Prüfsummen	3757
Arbeiten Sie mit Amazon S3 S3-Objektintegrität	3761
Mit versionierten Objekten arbeiten	3792
Serverless-Beispiele	3799
Aufrufen einer Lambda-Funktion über einen Amazon-S3-Auslöser	3799
Fehlerbehebung	3812
Beheben Sie den verweigerten Zugriff (403 Forbidden) Fehler	3812
Beispiele für abgelehnte Nachrichten und deren Behebung	3814
Bucket-Richtlinien und IAM Richtlinien	3823
Amazon S3 ACL S3-Einstellungen	3826
S3-Block-Public-Access-Einstellungen	3829
Amazon-S3-Verschlüsselungseinstellungen	3829
S3-Einstellungen für die Objektsperre	3831
VPC-Endpunktrichtlinien	3833
AWS Organizations policies	3833
Zugriffspunkteinstellungen	3833
Fehlerbehebung bei Batch Operations	3834
Der Auftragsbericht wird nicht bereitgestellt, wenn ein Problem mit Berechtigungen besteht oder ein Aufbewahrungsmodus aktiviert ist.	3835
Batchreplikation fehlgeschlagen: Bei der Manifestgenerierung wurden keine Schlüssel gefunden, die den Filterkriterien entsprechen	3835
Fehler bei Batch Operations nach dem Hinzufügen einer neuen Replikationsregel	3836
Bei S3-Batch-Vorgängen scheitern Objekte mit dem Fehler 400 InvalidRequest	3836
Fehler bei Auftragserstellung mit aktiviertem Auftrags-Tagging	3837
Zugriff zum Lesen des Manifests verweigert	3837
Behebung von Problemen im Lebenszyklus	3838
Ich habe eine Auflistungsoperation für meinen Bucket ausgeführt und es wurden Objekte angezeigt, von denen ich dachte, dass sie abgelaufen oder aufgrund einer Lebenszyklusregel übergeben worden waren.	3839
Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?	3839
Die Anzahl meiner S3-Objekte steigt immer noch, auch wenn ich Lebenszyklusregeln für einen Bucket mit aktivierter Versionierung eingerichtet habe.	3840
Wie leere ich meinen S3-Bucket mithilfe von Lebenszyklusregeln?	3842
Meine Abrechnung für Amazon S3 weist nach der Übergabe von Objekten in eine kostengünstigere Speicherklasse höhere Kosten auf.	3842

Ich habe meine Bucket-Richtlinie aktualisiert, meine S3-Objekte werden jedoch noch immer aufgrund abgelaufener Lebenszyklusregeln gelöscht.	3843
Kann ich S3-Objekte wiederherstellen, die aufgrund von S3-Lebenszyklusregeln abgelaufen sind?	3844
Wie kann ich ein Präfix aus meiner Lebenszyklusregel ausschließen?	3844
Wie kann ich mehrere Präfixe in meine Lebenszyklusregel aufnehmen?	3844
Fehlerbehebung bei der Replikation	3845
Tipps zur Fehlerbehebung in S3 Replication	3845
Fehler bei der Batchreplikation	3852
Behebung von Fehlern bei der Server-Zugriffsprotokollierung	3853
Häufige Fehlermeldungen beim Einrichten der Protokollierung	3853
Behebung von Bereitstellungsfehlern	3854
Fehlerbehebung für die Versionsverwaltung	3856
Ich möchte Objekte wiederherstellen, die in einem Bucket mit aktivierter Versionsverwaltung versehentlich gelöscht wurden.	3856
Ich möchte versionierte Objekte dauerhaft löschen	3858
Nach dem Aktivieren der Bucket-Versionsverwaltung stelle ich Leistungseinbußen fest	3859
Holen Sie sich eine Amazon S3 S3-Anfrage IDs für AWS Support	3861
Wird verwendet HTTP, um eine Anfrage zu erhalten IDs	3862
Verwenden Sie einen Webbrowser, um eine Anfrage zu erhalten IDs	3862
Verwenden Sie die AWS SDKs, um eine Anfrage zu erhalten IDs	3863
Verwenden von, um AWS CLI eine Anfrage zu erhalten IDs	3865
Verwenden Sie Windows PowerShell , um die Anfrage abzurufen IDs	3865
Verwenden von AWS CloudTrail Datenereignissen zum Abrufen von Anfragen IDs	3865
Verwenden Sie die S3-Serverzugriffsprotokollierung, um eine Anfrage abzurufen IDs	3866
Dokumentverlauf	3867
Frühere Updates	3907
AWS-Glossar	3936
.....	mmmcmmxxxvii

Was ist Amazon S3?

Amazon Simple Storage Service (Amazon S3) ist ein Objektspeicherservice, der branchenführende Skalierbarkeit, Datenverfügbarkeit, Sicherheit und Leistung bietet. Kunden aller Größen und Branchen können Amazon S3 für die Speicherung und den Schutz beliebiger Datenmengen für eine Reihe von Anwendungsfällen verwenden, wie Data Lakes, Websites, mobile Anwendungen, Backup und Wiederherstellung, Archivierung, Unternehmensanwendungen, IoT-Geräte und Big-Data-Analysen. Amazon S3 bietet Verwaltungsfunktionen, mit denen Sie den Zugriff auf Ihre Daten optimieren, organisieren und konfigurieren können, um Ihre spezifischen geschäftlichen, organisatorischen und Compliance-Anforderungen zu erfüllen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Funktionen von Amazon S3](#)
- [Funktionsweise von Amazon S3](#)
- [Amazon S3-Datenkonsistenzmodell](#)
- [Zugehörige Services](#)
- [Zugriff auf Amazon S3](#)
- [Bezahlen für Amazon S3](#)
- [PCIDSS-Einhaltung der Vorschriften](#)

Funktionen von Amazon S3

Speicherklassen

Amazon S3 bietet eine Vielzahl von Speicherklassen an, die für diverse Anwendungsfälle genutzt werden können. So können Sie beispielsweise missionskritische Produktionsdaten in S3 Standard für häufigen Zugriff speichern, Kosten sparen, indem Sie selten aufgerufene Daten in S3 Standard-

IA speichern und Daten zu den niedrigsten Kosten in S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive archivieren.

Amazon S3 Express One Zone ist eine leistungsstarke Amazon-S3-Speicherklasse mit einer einzelnen Zone, die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die derzeit verfügbare Cloud-Objektspeicherklasse mit der niedrigsten Latenz. Sie bietet Datenzugriffsgeschwindigkeiten, die bis zu zehnmals schneller sind und die Anforderungskosten um 50 Prozent niedriger sind als bei S3 Standard. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Um die Zugriffsgeschwindigkeit weiter zu erhöhen und Hunderttausende von Anfragen pro Sekunde zu unterstützen, werden Daten außerdem in einem neuen Bucket-Typ gespeichert: einem Amazon-S3-Verzeichnis-Bucket. Weitere Informationen finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Sie können Daten mit sich ändernden oder unbekanntem Zugriffsmustern in S3 Intelligent-Tiering speichern, das die Speicherkosten optimiert, indem es Ihre Daten automatisch zwischen vier Zugriffsebenen verschiebt, wenn sich Ihre Zugriffsmuster ändern. Diese vier Zugriffsebenen umfassen zwei Zugriffsebenen mit geringer Latenz, die für häufigen und seltenen Zugriff optimiert sind, und zwei Opt-in-Archivzugriffsebenen, die für den asynchronen Zugriff auf selten abgerufene Daten ausgelegt sind.

Weitere Informationen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Speicherverwaltung

Amazon S3 hat Speicherverwaltungsfunktionen, die Sie benutzen können, um Kosten zu verwalten, gesetzliche Anforderungen zu erfüllen, Latenz zu reduzieren und mehrere verschiedene Kopien Ihrer Daten für Compliance-Anforderungen zu speichern.

- [S3-Lebenszyklus](#) – Richten Sie eine Lebenszykluskonfiguration ein, um Ihre Objekte zu verwalten und während ihres gesamten Lebenszyklus kostengünstig zu speichern. Sie können Objekte in andere S3-Speicherklassen umstellen oder Objekte ablaufen, die das Ende ihrer Lebensdauer erreichen.
- [S3-Objektsperre](#) – Verhindern Sie, dass Amazon S3-Objekte für einen bestimmten Zeitraum oder auf unbestimmte Zeit gelöscht oder überschrieben werden. Sie können Object Lock verwenden, um gesetzliche Anforderungen zu erfüllen, die Speicherplatz write-once-read-many(WORM)

erfordern, oder um einfach eine weitere Schutzebene gegen Objektänderungen und -löschungen hinzuzufügen.

- [S3-Replikation](#) — Replizieren Sie Objekte und ihre jeweiligen Metadaten und Objekt-Tags in einen oder mehrere Ziel-Buckets im gleichen oder unterschiedlichen Ziel-Buckets, AWS-Regionen um Latenz, Compliance, Sicherheit und andere Anwendungsfälle zu reduzieren.
- [S3 Batch Operations](#) — Verwalten Sie Milliarden von Objekten in großem Umfang mit einer einzigen API S3-Anfrage oder ein paar Klicks in der Amazon S3 S3-Konsole. Mithilfe von Batch-Operationen können Sie Operationen wie Kopieren, AWS Lambda-Funktion aufrufen und Wiederherstellen für Millionen oder Milliarden von Objekten ausführen.

Zugriffsverwaltung und Sicherheit

Amazon S3 bietet Funktionen für die Überwachung und Verwaltung des Zugriffs auf Ihre Buckets und Objekte. Standardmäßig werden S3-Buckets und -Objekte als privat eingestuft. Sie haben nur Zugriff auf die S3-Ressourcen, die Sie erstellen. Um detaillierte Ressourcenberechtigungen zu erteilen, die Ihren speziellen Anwendungsfall unterstützen, oder um die Berechtigungen Ihrer Amazon S3-Ressourcen zu überprüfen, können Sie die folgenden Funktionen verwenden.

- [S3 öffentlichen Zugriff blockieren](#) – Blockieren Sie den öffentlichen Zugriff auf S3-Buckets und -Objekte. Standardmäßig sind die Einstellungen für das Blockieren des öffentlichen Zugriffs auf Bucket-Ebene aktiviert. Es wird empfohlen, alle Einstellungen für das Blockieren des öffentlichen Zugriffs aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen finden Sie unter [Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets](#).
- [AWS Identity and Access Management \(IAM\)](#) — IAM ist ein Webservice, mit dem Sie den Zugriff auf AWS Ressourcen, einschließlich Ihrer Amazon S3 S3-Ressourcen, sicher kontrollieren können. Mit können Sie Berechtigungen zentral verwalten IAM, mit denen gesteuert wird, auf welche AWS Ressourcen Benutzer zugreifen können. Sie steuern IAM damit, wer authentifiziert (angemeldet) und autorisiert ist (über Berechtigungen verfügt), um Ressourcen zu verwenden.
- [Bucket-Richtlinien](#) — Verwenden Sie eine IAM basierte Richtlinienprache, um ressourcenbasierte Berechtigungen für Ihre S3-Buckets und die darin enthaltenen Objekte zu konfigurieren.
- [Amazon-S3-Zugriffspunkte](#) – Konfigurieren Sie benannte Netzwerkendpunkte mit dedizierten Zugriffsrichtlinien, um den Datenzugriff auf freigegebene Datensätze in Amazon S3 skalierbar zu verwalten.

- [Zugriffskontrolllisten \(ACLs\)](#) — Erteilen Sie autorisierten Benutzern Lese- und Schreibberechtigungen für einzelne Buckets und Objekte. In der Regel empfehlen wir, stattdessen ressourcenbasierte S3-Richtlinien (Bucket-Richtlinien und Zugriffspunktrichtlinien) oder IAM Benutzerrichtlinien für die Zugriffskontrolle zu verwenden. ACLs Richtlinien stellen eine vereinfachte und flexiblere Zugriffskontrolloption dar. Mit Bucket- und Zugriffspunktrichtlinien können Sie Regeln definieren, die allgemein für alle Anfragen an Ihre Amazon-S3-Ressourcen gelten. Weitere Informationen zu den spezifischen Fällen, in denen Sie ACLs anstelle von ressourcenbasierten Richtlinien oder IAM Benutzerrichtlinien verwenden würden, finden Sie unter [Verwaltung des Zugriffs mit ACLs](#)
- [S3 Object Ownership](#) – Übernehmen Sie die Verantwortung für jedes Objekt in Ihrem Bucket, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, die Sie zum Deaktivieren oder Aktivieren verwenden können. ACLs Standardmäßig ACLs sind sie deaktiviert. Wenn diese ACLs Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.
- [IAMAccess Analyzer for S3](#) — Evaluieren und überwachen Sie Ihre S3-Bucket-Zugriffsrichtlinien und stellen Sie sicher, dass die Richtlinien nur den beabsichtigten Zugriff auf Ihre S3-Ressourcen ermöglichen.

Datenverarbeitung

Um Daten zu transformieren und Workflows auszulösen, um eine Vielzahl anderer Verarbeitungsaktivitäten maßstabsgerecht zu automatisieren, können Sie die folgenden Funktionen verwenden.

- [S3 Object Lambda](#) — Fügen Sie Ihren eigenen Code zu S3 hinzu und HEAD LIST fordern Sie anGET, Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Filtern Sie Zeilen, ändern Sie die Größe von Bildern dynamisch, verkleinern Sie vertrauliche Daten und vieles mehr.
- [Ereignisbenachrichtigungen](#) — Löst Workflows aus, die Amazon Simple Notification Service (AmazonSNS), Amazon Simple Queue Service (AmazonSQS) verwenden, und AWS Lambda wenn eine Änderung an Ihren S3-Ressourcen vorgenommen wird.

Speicherprotokollierung und Überwachung

Amazon S3 bietet Protokollierungs- und Überwachungstools, mit denen Sie überwachen und steuern können, wie Ihre Amazon S3-Ressourcen verwendet werden. Weitere Informationen finden Sie unter [Überwachungstools](#).

Automatisierte Überwachungstools

- [CloudWatchAmazon-Metriken für Amazon S3](#) — Verfolgen Sie den Betriebsstatus Ihrer S3-Ressourcen und konfigurieren Sie Abrechnungswarnungen, wenn geschätzte Gebühren einen benutzerdefinierten Schwellenwert erreichen.
- [AWS CloudTrail](#) — Zeichnet Aktionen auf, die von einem Benutzer, einer Rolle oder einem AWS-Service in Amazon S3 ausgeführt wurden. CloudTrail Protokolle bieten Ihnen eine detaillierte API Nachverfolgung von S3-Vorgängen auf Bucket- und Objektebene.

Manuelle Überwachungstools

- [Server-Zugriffsprotokollierung](#) – Abrufen detaillierter Datensätze für die Anforderungen, die an einen Bucket gestellt werden. Sie können Serverzugriffsprotokolle für viele Anwendungsfälle verwenden, z. B. für die Durchführung von Sicherheits- und Zugriffsprüfungen, das Kennenlernen Ihres Kundenstamms und das Verstehen Ihrer Amazon S3-Rechnung.
- [AWS Trusted Advisor](#) — Evaluieren Sie Ihr Konto anhand von AWS Best-Practice-Checks, um Möglichkeiten zur Optimierung Ihrer AWS Infrastruktur, zur Verbesserung von Sicherheit und Leistung, zur Kostensenkung und zur Überwachung der Servicekontingenten zu ermitteln. Sie können dann die Empfehlungen befolgen, um Ihre Dienste und Ressourcen zu optimieren.

Analytik und Einblicke

Amazon S3 bietet Funktionen, die Ihnen dabei helfen, Einblicke in Ihre Speicherauslastung zu erhalten, sodass Sie Ihren Speicher in großem Umfang besser verstehen, analysieren und optimieren können.

- [Amazon S3 Storage Lens](#) – Verstehen, analysieren und optimieren Sie Ihren Speicher. S3 Storage Lens bietet mehr als 60 Nutzungs- und Aktivitätsmetriken sowie interaktive Dashboards, um Daten für Ihr gesamtes Unternehmen, bestimmte Konten AWS-Regionen, Buckets oder Präfixe zu aggregieren.

- [Speicherklassenanalyse](#) – Analysieren Sie Speicherzugriffsmuster, um zu entscheiden, wann es an der Zeit ist, Daten in eine kostengünstigere Speicherklasse zu verschieben.
- [S3-Inventory mit Bestandsberichten](#) – Überwachen und erstellen Sie Berichte zu Objekten und den entsprechenden Metadaten und konfigurieren Sie andere Amazon S3-Funktionen, um in Inventarberichten Maßnahmen zu ergreifen. Sie können beispielsweise einen Bericht über den Replikations- und Verschlüsselungsstatus Ihrer Objekte erstellen. Eine Liste aller verfügbaren Metadaten, die für jedes Objekt in Inventarberichten verfügbar sind, finden Sie unter [Amazon S3-Bestandsliste](#).

Starke Konsistenz

Amazon S3 bietet insgesamt eine hohe read-after-write Konsistenz für Objekte PUT und DELETE Anfragen von Objekten in Ihrem Amazon S3 S3-Bucket AWS-Regionen. Dieses Verhalten gilt sowohl für Schreibvorgänge neuer Objekte als auch für PUT Anfragen, die bestehende Objekte und DELETE Anfragen überschreiben. Darüber hinaus sind die Lesevorgänge auf Amazon S3 Select, Amazon S3 S3-Zugriffskontrolllisten (ACLs), Amazon S3 S3-Objekt-Tags und Objektmetadaten (z. B. dem HEAD Objekt) stark konsistent. Weitere Informationen finden Sie unter [Amazon S3-Datenkonsistenzmodell](#).

Funktionsweise von Amazon S3

Amazon S3 ist ein Objektspeicherdienst, der Daten als Objekte in Buckets speichert. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben. Ein Bucket ist ein Container für Objekte.

Um Ihre Daten in Amazon S3 zu speichern, erstellen Sie zunächst einen Bucket und geben einen Bucket-Namen und AWS-Region an. Anschließend laden Sie Ihre Daten in diesen Bucket als Objekte in Amazon S3 hoch. Jedes Objekt hat einen Schlüssel (oder Schlüsselnamen), der der eindeutige Bezeichner für das Objekt im Bucket ist.

S3 bietet Funktionen, die Sie konfigurieren können, um Ihren speziellen Anwendungsfall zu unterstützen. Beispielsweise können Sie das S3-Versioning verwenden, um mehrere Versionen eines Objekts im selben Bucket zu halten, wodurch Sie versehentlich gelöschte oder überschriebene Objekte wiederherstellen können.

Buckets und die darin enthaltenen Objekte sind privat und können nur zugegriffen werden, wenn Sie explizit Zugriffsberechtigungen erteilen. Sie können Bucket-Richtlinien, AWS Identity and Access Management (IAM) -Richtlinien, Zugriffskontrolllisten (ACLs) und S3-Zugangspunkte verwenden, um den Zugriff zu verwalten.

Themen

- [Buckets](#)
- [Objekte](#)
- [Schlüssel](#)
- [S3-Versioning](#)
- [Versions-ID](#)
- [Bucket-Richtlinie](#)
- [S3-Zugriffspunkte](#)
- [Zugriffskontrolllisten \(ACLs\)](#)
- [Regionen](#)

Buckets

Ein Bucket ist ein Behälter für Objekte, die in Amazon S3 gespeichert werden. Sie können beliebig viele Objekte in einem Bucket speichern und bis zu 100 Buckets in Ihrem Konto haben. Um eine Erhöhung anzufordern, rufen Sie die [Service-Quotas-Konsole](#) auf.

Jedes Objekt ist in einem Bucket enthalten. Wenn das benannte Objekt beispielsweise im amzn-s3-demo-bucket Bucket in der Region USA West (Oregon) gespeichert photos/puppy.jpg ist, kann es mithilfe von adressiert werden. URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg` Weitere Informationen siehe [Zugriff auf einen Bucket](#).

Wenn Sie einen Bucket erstellen, geben Sie einen Bucket-Namen ein und wählen die Option AWS-Region, wo sich der Bucket befindet. Nachdem Sie einen Bucket erstellt haben, können Sie den Namen des Buckets oder seiner Region nicht mehr ändern. Bucket-Namen müssen den [Regeln für die Benennung von Buckets](#) folgen. Sie können einen Bucket auch für die Verwendung des [S3-Versioning](#) oder anderer [Speicherverwaltungs](#)-Funktionen konfigurieren.

Zudem tun Buckets Folgendes:

- Sie strukturieren den Amazon S3-Namespaces auf der höchsten Ebene.
- Sie identifizieren das Konto, dem die Gebühren für Datenspeicherung und -übertragung belastet werden.

- Stellen Sie Zugriffskontrolloptionen wie Bucket-Richtlinien, Zugriffskontrolllisten (ACLs) und S3-Zugriffspunkte bereit, mit denen Sie den Zugriff auf Ihre Amazon S3 S3-Ressourcen verwalten können.
- Sie dienen im Rahmen der Erstellung von Nutzungsberichten als Auswertungseinheit.

Weitere Informationen über Buckets finden Sie unter [Bucket-Übersicht](#).

Objekte

Objekte sind die Grundeinheiten, die in Amazon S3 gespeichert werden. Objekte bestehen aus Objekt- und Metadaten. Metadaten bestehen aus mehreren Name/Wert-Paaren, die das Objekt beschreiben. Diese Paare enthalten einige Standardmetadaten, wie das Datum der letzten Änderung, und HTTP Standardmetadaten, wie Content-Type z. Sie können bei der Speicherung des Objekts auch benutzerdefinierte Metadaten angeben.

Ein Objekt wird innerhalb eines Buckets eindeutig durch einen [Schlüssel \(Name\)](#) und eine [Versions-ID](#) identifiziert (wenn das S3-Versioning im Bucket aktiviert ist). Weitere Informationen über Objekte finden Sie unter [Übersicht über Amazon-S3-Objekte](#).

Schlüssel

Ein Objektschlüssel (oder Schlüsselname) ist der eindeutige Bezeichner für ein Objekt in einem Bucket. Jedes Objekt in einem Bucket besitzt genau einen Schlüssel. Die Kombination aus Bucket, Objektschlüssel und optional Versions-ID (wenn das S3-Versioning für den Bucket aktiviert ist) identifiziert jedes Objekt eindeutig. Amazon S3 fungiert also als grundlegende Datenzuordnung zwischen „Bucket + Schlüssel + Version“ und dem Objekt selbst.

Jedes Objekt in Amazon S3 ist über eine Kombination von Webservice-Endpunkt, Bucket-Name, Schlüssel und wahlweise einer Version aufrufbar. In der *amzn-s3-demo-bucket* steht URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg` beispielsweise der Name des Buckets und `photos/puppy.jpg` ist der Schlüssel.

Weitere Informationen über Objektschlüssel finden Sie unter [Erstellen von Objektschlüsselnamen](#).

S3-Versioning

Sie S3-Versioning verwenden, um mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Mit S3-Versioning können Sie jede Version jedes in Ihren Buckets gespeicherten

Objekts beibehalten, abrufen und wiederherstellen. Sie können sowohl nach unbeabsichtigten Benutzeraktionen als auch nach Anwendungsfehlern problemlos wiederherstellen.

Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Versions-ID

Wenn Sie das S3-Versioning in einem Bucket aktivieren, generiert Amazon S3 eine eindeutige Versions-ID für jedes Objekt, das dem Bucket hinzugefügt wird. Objekte, die zum Zeitpunkt der Aktivierung des Versioning bereits im Bucket vorhanden waren, haben die Versions-ID null. Wenn Sie diese (oder andere) Objekte mit anderen Operationen wie [CopyObject](#) und ändern [PutObject](#), erhalten die neuen Objekte eine eindeutige Versions-ID.

Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Bucket-Richtlinie

Eine Bucket-Richtlinie ist eine ressourcenbasierte AWS Identity and Access Management (IAM) Richtlinie, mit der Sie Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte gewähren können. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt.

Bucket-Richtlinien verwenden eine auf der JSON Grundlage von Zugriffsrichtlinien verwendete Sprache, die in allen Bereichen Standard ist. AWS Sie können Bucket-Richtlinien verwenden, um Berechtigungen für die Objekte in einem Bucket hinzuzufügen oder zu verweigern. Bucket-Richtlinien erlauben oder verweigern Anforderungen basierend auf den Elementen in der Richtlinie, einschließlich des Anforderers, S3-Aktionen, Ressourcen und Aspekten oder Bedingungen der Anforderung (z. B. die IP-Adresse, die für die Anforderung verwendet wird). Sie können beispielsweise eine Bucket-Richtlinie erstellen, die kontoübergreifende Berechtigungen zum Hochladen von Objekten in einen S3-Bucket gewährt, während gleichzeitig sichergestellt wird, dass der Bucket-Eigentümer die volle Kontrolle über die hochgeladenen Objekte hat. Weitere Informationen finden Sie unter [Beispiele für Amazon S3 S3-Bucket-Richtlinien](#).

In Ihrer Bucket-Richtlinie können Sie Platzhalterzeichen für Amazon-Ressourcennamen (ARNs) und andere Werte verwenden, um einer Teilmenge von Objekten Berechtigungen zu erteilen. Sie können beispielsweise den Zugriff auf Gruppen von Objekten steuern, die mit einem gemeinsamen [Präfix](#) beginnen oder mit einer bestimmten Erweiterung wie `.html` enden.

S3-Zugriffspunkte

Amazon S3-Zugriffspunkte sind benannte Netzwerkendpunkte mit dedizierten Zugriffsrichtlinien, die beschreiben, wie mit diesem Endpunkt auf Daten zugegriffen werden kann. Access Points sind an Buckets angehängt, mit denen Sie S3-Objektoperationen wie und ausführen können. GetObject PutObject Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3.

Jeder Zugriffspunkt verfügt über eine eigene Zugriffspunktrichtlinie. Sie können [Block Public Access](#)-Einstellungen für jeden Zugriffspunkt konfigurieren. Um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk zu beschränken, können Sie auch jeden Access Point so konfigurieren, dass er nur Anfragen von einer virtuellen privaten Cloud akzeptiert (VPC).

Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Zugriffskontrolllisten (ACLs)

Sie können ACLs verwenden, um autorisierten Benutzern Lese- und Schreibberechtigungen für einzelne Buckets und Objekte zu gewähren. Jedem Bucket und Objekt ist eine Unterressource ACL angehängt. Das ACL definiert, welche AWS-Konten Gruppen Zugriff erhalten und welche Art von Zugriff gewährt wird. ACLs sind ein Zugangskontrollmechanismus, der älter ist als IAM. Mehr über ACLs erfahren Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Regionen

Sie können das geografische Gebiet wählen AWS-Region , in dem Amazon S3 die von Ihnen erstellten Buckets speichert. Sie sollten eine Region im Hinblick auf Latenz, Kosten sowie Einhaltung der relevanten Vorschriften auswählen. Objekte, die in der Region gespeichert sind und diese AWS-Region nie verlassen, es sei denn, Sie übertragen sie explizit in eine andere Region oder replizieren sie. In der Region EU (Irland) gespeicherte Objekte verlassen diese Region nicht.

Note

Sie können auf Amazon S3 und seine Funktionen nur in den Bereichen zugreifen AWS-Regionen , die für Ihr Konto aktiviert sind. Weitere Informationen darüber, wie Sie einer Region die Erstellung und Verwaltung von AWS Ressourcen ermöglichen, finden Sie unter [Verwaltung AWS-Regionen](#) in der Allgemeine AWS-Referenz.

Eine Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

Amazon S3-Datenkonsistenzmodell

Amazon S3 bietet insgesamt eine hohe read-after-write Konsistenz für Objekte PUT und DELETE Anfragen von Objekten in Ihrem Amazon S3 S3-Bucket AWS-Regionen. Dieses Verhalten gilt sowohl für Schreibvorgänge in neue Objekte als auch für PUT Anfragen, die bestehende Objekte und DELETE Anfragen überschreiben. Darüber hinaus sind die Lesevorgänge auf Amazon S3 Select, Amazon S3 S3-Zugriffskontrolllisten (ACLs), Amazon S3 S3-Objekt-Tags und Objektmetadaten (z. B. das HEAD Objekt) sehr konsistent.

Aktualisierungen an einem einzelnen Schlüssel sind unteilbar. Wenn Sie beispielsweise von einem Thread aus eine PUT Anfrage an einen vorhandenen Schlüssel stellen und gleichzeitig von einem zweiten Thread aus eine GET Anfrage für denselben Schlüssel ausführen, erhalten Sie entweder die alten oder die neuen Daten, jedoch niemals unvollständige oder beschädigte Daten.

Amazon S3 erzielt hohe Verfügbarkeit, indem die Daten innerhalb der AWS -Rechenzentren über mehrere Server repliziert werden. Wenn eine PUT Anfrage erfolgreich ist, werden Ihre Daten sicher gespeichert. Bei jedem GET LIST Lesevorgang (oder jeder Anfrage), der nach Erhalt einer erfolgreichen PUT Antwort eingeleitet wird, werden die in der PUT Anfrage geschriebenen Daten zurückgegeben. Hier finden Sie Beispiele für dieses Verhalten:

- Ein Prozess schreibt ein neues Objekt in Amazon S3 und listet sofort Schlüssel innerhalb seines Buckets auf. Das neue Objekt wird in der Liste angezeigt.
- Ein Prozess ersetzt ein vorhandenes Objekt und versucht sofort, es zu lesen. Amazon S3 gibt die neuen Daten zurück.
- Ein Prozess löscht ein vorhandenes Objekt und versucht sofort, es zu lesen. Amazon S3 gibt keine Daten zurück, da das Objekt gelöscht wurde.
- Ein Prozess löscht ein vorhandenes Objekt und listet sofort Schlüssel innerhalb seines Buckets auf. Das Objekt wird nicht im Verzeichnis angezeigt.

Note

- Amazon S3 unterstützt keine Objektsperre für gleichzeitige Schreibvorgänge. Wenn zwei PUT Anfragen gleichzeitig an denselben Schlüssel gestellt werden, gewinnt die Anfrage mit dem neuesten Zeitstempel. Wenn das ein Problem ist, müssen Sie in Ihre Anwendung einen Sperrmechanismus für Objekte einbauen.
- Updates basieren auf Schlüsseln. Es gibt keine Möglichkeit für unteilbare Aktualisierungen über Schlüssel hinweg. Sie können beispielsweise die Aktualisierung eines Schlüssels nicht von der Aktualisierung eines anderen Schlüssels abhängig machen. Dazu müssten Sie diese Funktionalität in Ihrer Anwendung implementieren.

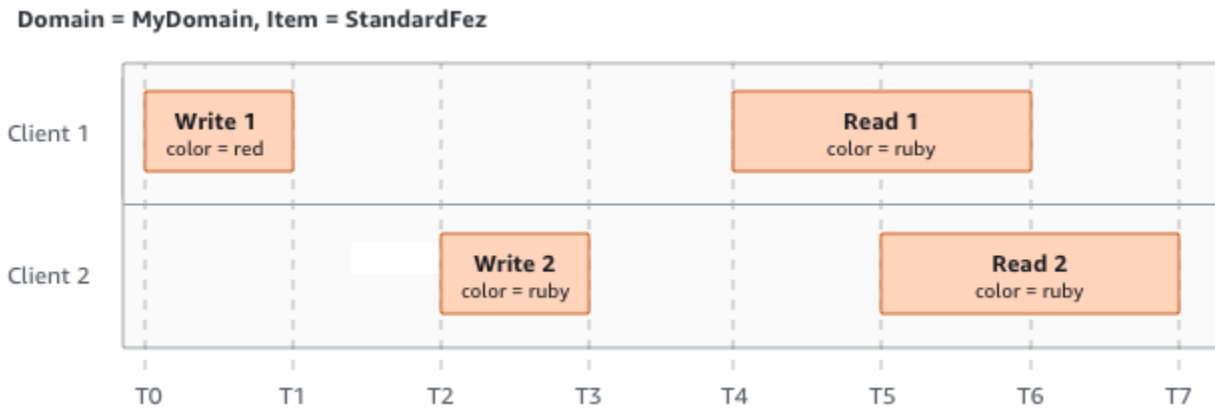
Bucket-Konfigurationen haben ein Modell mit letztendlicher Datenkonsistenz. Dies bedeutet insbesondere, dass:

- Wenn Sie einen Bucket löschen und sofort alle Buckets auflisten, wird der gelöschte Bucket möglicherweise weiterhin in der Liste angezeigt.
- Wenn Sie das Versioning für einen Bucket zum ersten Mal aktivieren, kann es einen Moment dauern, bis die Änderung vollständig verbreitet ist. Wir empfehlen, nach der Aktivierung der Versionierung 15 Minuten zu warten, bevor Sie Schreibvorgänge (PUT oder DELETE Anfragen) für Objekte im Bucket ausführen.

Gleichzeitige Anwendungen

Dieser Abschnitt enthält Beispiele für das zu erwartende Verhalten von Amazon S3, wenn mehrere Kunden in dieselben Elemente schreiben.

In diesem Beispiel werden W1 (Lesen 1) und W2 (Lesen 2) abgeschlossen, bevor R1 (Lesen 1) und R2 (Lesen 2) starten. Da S3 stark konsistent ist, geben R1 und R2 beide `color = ruby` zurück.

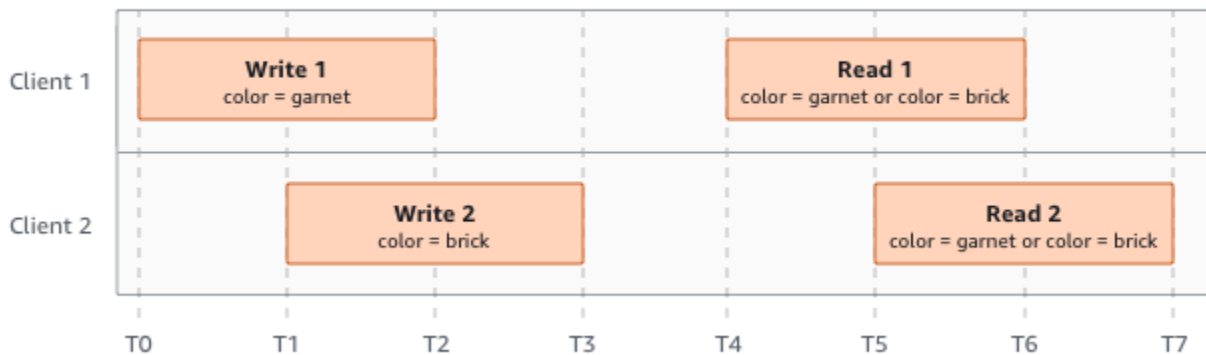


Im nächsten Beispiel ist W2 vor dem Start von R1 nicht abgeschlossen. Daher könnte R1 `color = ruby` oder `color = garnet` zurückgeben. Da W1 und W2 jedoch vor dem Start von R2 fertig sind, gibt R2 zurück `color = garnet`.



Im letzten Beispiel beginnt W2, bevor W1 eine Bestätigung erhalten hat. Daher werden diese Schreibvorgänge als gleichzeitig betrachtet. Amazon S3 verwendet intern last-writer-wins Semantik, um zu bestimmen, welcher Schreibvorgang Vorrang hat. Die Reihenfolge, in der Amazon S3 die Anforderungen erhält, und die Reihenfolge, in der Anwendungen Bestätigungen erhalten, können aber aufgrund von verschiedenen Faktoren wie Netzwerklatenz nicht vorhergesagt werden. Beispielsweise kann W2 von einer EC2 Amazon-Instance in derselben Region initiiert werden, während W1 von einem weiter entfernten Host initiiert werden kann. Die beste Methode, um den endgültigen Wert zu bestimmen, besteht darin, einen Lesevorgang auszuführen, nachdem beide Schreibvorgänge bestätigt wurden.

Domain = MyDomain, Item = StandardFez



Zugehörige Services

Nachdem Sie Ihre Daten in Amazon S3 geladen haben, können Sie sie mit anderen AWS Diensten verwenden. Häufig genutzte Services:

- [Amazon Elastic Compute Cloud \(AmazonEC2\)](#) — Bietet sichere und skalierbare Rechenkapazität in der AWS Cloud. EC2 Durch die Nutzung von Amazon müssen Sie nicht im Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwenden EC2, um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten.
- [Amazon EMR](#) — Unterstützt Unternehmen, Forscher, Datenanalysten und Entwickler bei der einfachen und kostengünstigen Verarbeitung großer Datenmengen. Amazon EMR verwendet ein gehostetes Hadoop-Framework, das auf der Web-Scale-Infrastruktur von Amazon EC2 und Amazon S3 läuft.
- [AWS Snow Family](#) — Hilft Kunden, die Betriebsabläufe in rauen Umgebungen außerhalb von Rechenzentren und an Standorten durchführen müssen, an denen es an konsistenter Netzwerkkonnektivität mangelt. Sie können AWS Snow Family-Geräte verwenden, um lokal und kostengünstig auf die Speicher- und Rechenleistung von AWS Cloud Orten zuzugreifen, an denen eine Internetverbindung möglicherweise keine Option ist.
- [AWS Transfer Family](#) — Bietet vollständig verwaltete Unterstützung für Dateiübertragungen direkt in und aus Amazon S3 oder Amazon Elastic File System (AmazonEFS) mithilfe von Secure Shell (SSH) File Transfer Protocol (SFTP), File Transfer Protocol over SSL (FTPS) und File Transfer Protocol (FTP).

Zugriff auf Amazon S3

Sie können mit Amazon S3 auf eine der folgenden Arten arbeiten:

AWS Management Console

Die Konsole ist eine webbasierte Benutzeroberfläche für die Verwaltung von Amazon S3 und AWS Ressourcen. Wenn Sie sich für eine angemeldet haben AWS-Konto, können Sie auf die Amazon S3 S3-Konsole zugreifen, indem Sie sich bei der anmelden AWS Management Console und auf der AWS Management Console Startseite S3 auswählen.

AWS Command Line Interface

Sie können die AWS Befehlszeilentools verwenden, um Befehle auszugeben oder Skripts an der Befehlszeile Ihres Systems zu erstellen, um Aufgaben AWS (einschließlich S3) auszuführen.

Das [AWS Command Line Interface \(AWS CLI\)](#) bietet Befehle für eine Vielzahl von AWS-Services. Das AWS CLI wird unter Windows, MacOS und Linux unterstützt. Informationen zu den ersten Schritten finden Sie im [AWS Command Line Interface -Benutzerhandbuch](#). Weitere Informationen über die Befehle für Amazon S3 finden Sie unter [s3api](#) und [s3control](#) in der AWS CLI -Befehlsreferenz.

AWS SDKs

AWS stellt SDKs (Software Development Kits) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen bestehen (Java, Python, Ruby, .NET, iOS, Android usw.). AWS SDKs Sie bieten eine bequeme Möglichkeit, programmatischen Zugriff auf S3 und AWS zu erstellen. Amazon S3 ist ein REST Service. Sie können Anfragen mithilfe der AWS SDK Bibliotheken an Amazon S3 senden, die das zugrunde liegende Amazon S3 einschließen REST API und Ihre Programmieraufgaben vereinfachen. SDKs Sie kümmern sich beispielsweise um Aufgaben wie das Berechnen von Signaturen, das kryptografische Signieren von Anfragen, das Verwalten von Fehlern und das automatische Wiederholen von Anfragen. [Informationen zu den AWS SDKs, einschließlich deren Herunterladen und Installieren, finden Sie unter Tools für. AWS](#)

Jede Interaktion mit Amazon S3 erfolgt entweder authentifiziert oder anonym. Wenn Sie die verwenden AWS SDKs, berechnen die Bibliotheken die Signatur für die Authentifizierung anhand der von Ihnen bereitgestellten Schlüssel. Weitere Informationen darüber, wie Sie Anforderungen an Amazon S3 stellen, finden Sie unter [Senden von Anforderungen](#).

Amazon S3 REST API

Die Amazon S3-Architektur ist so ausgelegt, dass sie unabhängig von Programmiersprachen ist und unsere von AWS unterstützten Schnittstellen verwendet, um Objekte zu speichern und abzurufen. Sie können mithilfe von Amazon S3 und AWS programmatisch auf S3 zugreifen. REST API Das REST API ist eine HTTP Schnittstelle zu Amazon S3. Mit dem verwenden Sie HTTP Standardanforderungen RESTAPI, um Buckets und Objekte zu erstellen, abzurufen und zu löschen.

Um das zu verwenden RESTAPI, können Sie jedes Toolkit verwenden, das dies unterstützt. HTTP Sie können sogar einen Browser verwenden, um Objekte zu laden, wenn diese anonym lesbar sind.

Das REST API verwendet HTTP Standardheader und Statuscodes, sodass Standardbrowser und Toolkits erwartungsgemäß funktionieren. In einigen Bereichen haben wir Funktionen hinzugefügt HTTP (zum Beispiel haben wir Header hinzugefügt, um die Zugriffskontrolle zu unterstützen). In diesen Fällen haben wir unser Bestes getan, um die neuen Funktionen so hinzuzufügen, dass sie dem Stil der HTTP Standardnutzung entsprechen.

Wenn Sie in Ihrer Anwendung direkte REST API Anrufe tätigen, müssen Sie den Code zur Berechnung der Signatur schreiben und der Anfrage hinzufügen. Weitere Informationen darüber, wie Sie Anforderungen an Amazon S3 stellen, finden Sie unter [Senden von Anforderungen](#).

Note

SOAPAPISupport Over HTTP ist veraltet, aber es ist immer noch verfügbar. HTTPS Neuere Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Wir empfehlen Ihnen, entweder den REST API oder den zu verwenden AWS SDKs.

Bezahlen für Amazon S3

Die Preise für Amazon S3 sind so ausgelegt, dass Sie nicht im Hinblick auf die Speicheranforderungen Ihrer Anwendung planen müssen. Die meisten Speicheranbieter verlangen, dass Sie eine vorgegebene Menge an Speicherkapazität und Netzwerkübertragungskapazität erwerben. Wenn Sie in diesem Szenario diese Kapazität überschreiten, wird Ihr Service abgeschaltet, oder Sie zahlen hohe Überziehungsgebühren. Wenn Sie diese Kapazität nicht überschreiten, zahlen Sie genauso viel, als wenn Sie sie verwendet hätten.

Amazon S3 stellt Ihnen nur Gebühren für Kapazitäten in Rechnung, die Sie tatsächlich genutzt haben, ohne verborgene Kosten und Überziehungsgebühren. Dieses Modell bietet Ihnen einen

Service mit variablen Kosten, der mit Ihrem Unternehmen mitwachsen kann und Ihnen gleichzeitig die Kostenvorteile der Infrastruktur bietet. AWS Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Wenn Sie sich für registrieren AWS, AWS-Konto ist Ihr automatisch für alle Services in angemeldet AWS, einschließlich Amazon S3. Es werden jedoch nur die Services in Rechnung gestellt, die Sie tatsächlich nutzen. Wenn Sie neuer Amazon S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter [Kostenloses Kontingent für AWS](#).

Um Ihre Rechnung anzuzeigen, navigieren Sie zu Fakturierungs- und Kostenverwaltungs-Dashboard in der [AWS Billing and Cost Management -Konsole](#). Weitere Informationen zur AWS-Konto Abrechnung finden Sie im [AWS Billing Benutzerhandbuch](#). Wenn Sie Fragen zur AWS Abrechnung und haben AWS-Konten, wenden Sie sich an den [AWS Support](#).

PCIDSSEinhaltung der Vorschriften

Amazon S3 unterstützt die Verarbeitung, Speicherung und Übertragung von Kreditkartendaten durch einen Händler oder Dienstleister und wurde als konform mit dem Payment Card Industry (PCI) Data Security Standard (DSS) validiert. Weitere Informationen PCI DSS zum Compliance-Paket sowie zur Beantragung einer Kopie des AWS PCI Compliance-Pakets finden Sie unter [PCIDSSStufe 1](#).

Erste Schritte mit Amazon S3

Sie können mit Amazon S3 beginnen, indem Sie mit Buckets und Objekten arbeiten. Ein Bucket ist ein Container für Objekte. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben.

Um ein Objekt in Amazon S3 zu speichern, erstellen Sie einen Bucket und laden das Objekt dann in den Bucket hoch. Wenn sich das Objekt im Bucket befindet, können Sie es öffnen, herunterladen und verschieben. Wenn Sie kein Objekt oder einen Bucket mehr benötigen, können Sie Ihre Ressourcen aufräumen.

Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon S3](#). Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter [Kostenloses Kontingent für AWS](#).

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Video: Erste Schritte mit Amazon S3

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in [Voraussetzung: Einrichten von Amazon S3](#) beschriebenen Schritte ausgeführt haben.

Themen

- [Voraussetzung: Einrichten von Amazon S3](#)
- [Schritt 1: Erstellen des ersten S3-Buckets](#)
- [Schritt 2: Hochladen eines Objekts in Ihren Bucket](#)
- [Schritt 3: Herunterladen eines Objekts](#)
- [Schritt 4: Kopieren Ihres Objekts in einen Ordner](#)
- [Schritt 5: Löschen Ihrer Objekte und Buckets](#)
- [Nächste Schritte](#)

Voraussetzung: Einrichten von Amazon S3

Wenn Sie sich anmelden für AWS, dein AWS-Konto ist automatisch für alle Dienste angemeldet in AWS, einschließlich Amazon S3. Berechnet werden Ihnen aber nur die Services, die Sie nutzen.

Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon S3](#). Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter [AWS Kostenloses Kontingent](#).

Um Amazon S3 einzurichten, führen Sie die Schritte in den folgenden Abschnitten aus.

Wenn Sie sich anmelden für AWS und Amazon S3 einrichten, können Sie optional die Anzeigesprache in der AWS Management Console. Weitere Informationen finden Sie unter [Ändern der Sprache des AWS Management Console](#) in der AWS Management Console Handbuch „Erste Schritte“.

Themen

- [Melde dich an für ein AWS-Konto](#)
- [Erstellen eines Benutzers mit Administratorzugriff](#)

Melde dich an für ein AWS-Konto

Wenn Sie kein haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie [https://portal.aws.amazon.com/billing/die Anmeldung](https://portal.aws.amazon.com/billing/die-Anmeldung).
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Tasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, ein Root-Benutzer des AWS-Kontos wird erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen im Konto. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff erfordern](#).

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu <https://aws.amazon.com/> gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie sich Ihre Root-Benutzer des AWS-Kontos, aktivieren AWS IAM Identity Center, und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melde dich an bei [AWS Management Console](#) als Kontoinhaber wählen Sie Root-Benutzer und geben Sie Ihren AWS-Konto E-Mail-Adresse. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Als Root-Benutzer anmelden in der AWS-Anmeldung Benutzerleitfaden](#).

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für Ihren Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren Sie ein virtuelles MFA Gerät für Ihr AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) in der AWS IAM Identity Center Benutzerleitfaden.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Für ein Tutorial zur Verwendung des IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) in der AWS IAM Identity Center Benutzerleitfaden.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM Identity Center-Benutzer anzumelden, verwenden Sie die Anmeldung, URL die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM Identity Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie unter [Anmelden bei AWS Zugriffsportal](#) im AWS-Anmeldung Benutzerleitfaden.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie unter [Einen Berechtigungssatz erstellen in](#) der AWS IAM Identity Center Benutzerleitfaden.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Anweisungen finden [Sie unter Gruppen hinzufügen](#) im AWS IAM Identity Center Benutzerleitfaden.

Schritt 1: Erstellen des ersten S3-Buckets

Nachdem Sie sich angemeldet haben für AWS, Sie sind bereit, einen Bucket in Amazon S3 mit dem zu erstellen AWS Management Console. Jedes Objekt in Amazon S3 wird in einem Bucket gespeichert. Bevor Sie Daten in Amazon S3 speichern können, müssen Sie einen Bucket erstellen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Note

Für das Erstellen eines Buckets werden Ihnen keine Gebühren in Rechnung gestellt. Es fallen nur Gebühren für das Speichern von Objekten im Bucket und für die Übertragung von Objekten in den und aus dem Bucket an. Die Gebühren, die beim Durcharbeiten der Beispiele in diesem Handbuch anfallen, sind minimal (weniger als 1,0 USD). Weitere Information zu Speicherkosten finden Sie unter [Amazon S3 – Preise](#).

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

 Note

Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Für eine Liste von Amazon S3 AWS-Regionen, siehe [AWS-Service Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.


Anschließend wird die Seite Bucket erstellen geöffnet.

5. Sehen Sie sich unter Allgemeine Konfiguration den AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Allgemein aus.
7. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

Der Bucket-Name ...:


- Muss innerhalb einer Partition einzigartig sein. Eine Partition ist eine Gruppe von Regionen. AWS hat derzeit drei Partitionen: aws (Standardregionen), aws-cn (Regionen in China) und aws-us-gov (AWS GovCloud (US) Regions).
- zwischen 3 und 63 Zeichen lang sein,
- Darf nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen. Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Das Tool AWS-Konto wer den Bucket erstellt, besitzt ihn. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

 **Important**

Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name ist in dem Bereich sichtbarURLs, der auf die Objekte im Bucket verweist.

8. AWS Management Console ermöglicht es Ihnen, die Einstellungen eines vorhandenen Buckets in Ihren neuen Bucket zu kopieren. Wenn Sie die Einstellungen eines vorhandenen Buckets nicht kopieren möchten, fahren Sie mit dem nächsten Schritt fort.

 **Note**

Diese Option:

- Ist nicht verfügbar in der AWS CLI und ist nur in der Konsole verfügbar
- Ist für Verzeichnis-Buckets nicht verfügbar
- Kopiert die Bucket-Richtlinie nicht aus dem vorhandenen Bucket in den neuen Bucket

Um die Einstellungen eines vorhandenen Buckets zu kopieren, wählen Sie unter Einstellungen aus vorhandenem Bucket kopieren die Option Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geöffnet. Suchen Sie den Bucket mit den Einstellungen, die Sie kopieren möchten, und wählen Sie Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geschlossen, und das Fenster „Bucket erstellen“ wird erneut geöffnet.

Unter Einstellungen aus vorhandenem Bucket kopieren wird nun der Name des ausgewählten Buckets angezeigt. Außerdem wird die Option Standardwerte wiederherstellen angezeigt, mit der Sie die kopierten Bucket-Einstellungen entfernen können. Überprüfen Sie die verbleibenden Bucket-Einstellungen auf der Seite Bucket erstellen. Sie werden sehen, dass sie jetzt mit den Einstellungen des von Ihnen ausgewählten Buckets übereinstimmen. Sie können mit dem letzten Schritt fortfahren.

9. Wählen Sie unter Object Ownership eine der folgenden Einstellungen aus, um die Inhaberschaft von Objekten, die in Ihren Bucket hochgeladen wurden, zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

ACLs deaktiviert

- Bucket Owner enforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLs wirken sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

ACLs sind standardmäßig deaktiviert. Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option weiterhin zu deaktivieren, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

ACLs aktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreiben ACL.

Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control` ACL, können Sie [eine Bucket-Richtlinie hinzufügen](#), die nur Objekt-Uploads zulässt, die dies verwenden. ACL


- Object Writer — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Note

Die Standardeinstellung ist Bucket-Eigentümer erzwungen. Um die Standardeinstellung anzuwenden und ACLs deaktiviert zu lassen, ist nur die `s3:CreateBucket` entsprechende Berechtigung erforderlich. Zum Aktivieren ACLs benötigen Sie die `s3:PutBucketOwnershipControls` entsprechende Genehmigung.

10. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.

Alle vier Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

 Note

Zum Aktivieren aller Einstellungen zum Blockieren des öffentlichen Zugriffs ist nur die `s3:CreateBucket`-Berechtigung erforderlich. Wenn Sie eine der Einstellungen zum Blockieren des öffentlichen Zugriffs deaktivieren möchten, benötigen Sie die `s3:PutBucketPublicAccessBlock`-Berechtigung.

11. (Optional) Unter Bucket Versioning (Bucket-Versionsverwaltung) können Sie auswählen, ob Sie Varianten von Objekten in Ihrem Bucket beibehalten möchten. Weitere Informationen über das Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).


Wenn Sie die Versionsverwaltung in Ihrem Bucket deaktivieren oder aktivieren möchten, wählen Sie entweder Disable (Deaktivieren) oder Enable (Aktivieren) aus.

12. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher verwendet werden.

Wenn Sie ein Bucket-Tag hinzuzufügen, geben Sie einen Key (Schlüssel) und optional einen Value (Wert) ein. Wählen Sie dann Add Tag (Tag hinzufügen) aus.

13. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
14. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Standardverschlüsselung zu konfigurieren:

- Verwalteter Amazon S3 S3-Schlüssel (SSE-S3)
- AWS Key Management Service Schlüssel (SSE-KMS)

 Important

Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, unterliegen Sie dem Kontingent für Anfragen pro Sekunde (RPS) von AWS KMS. Für weitere Informationen über AWS

KMS Kontingente und wie Sie eine Quotenerhöhung beantragen können, finden Sie unter [Kontingente](#) in der AWS Key Management Service Leitfaden für Entwickler.

Buckets und neue Objekte werden mit serverseitiger Verschlüsselung verschlüsselt. Dabei ist ein Von Amazon S3 verwalteter Schlüssel die Grundebene der Verschlüsselungskonfiguration. Weitere Informationen zur Standardverschlüsselung finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

15. Wenn du wählst AWS Key Management Service Taste (SSE-KMS), gehen Sie wie folgt vor:
 - a. Unter AWS KMS key, geben Sie Ihren KMS Schlüssel auf eine der folgenden Arten an:
 - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie Ihren KMS Schlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

 - Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
 - Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen, klicken Sie auf AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

 **Important**

Sie können nur KMS Schlüssel verwenden, die in derselben Version verfügbar sind AWS-Region wie der Eimer. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS

Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [KMSSchlüssel erstellen, die andere Konten verwenden können](#), in AWS Key Management Service Leitfaden für Entwickler. Weitere Informationen zu SSE - finden KMS Sie unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#).

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter [Identifizieren symmetrischer und KMS asymmetrischer Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.


Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler. Weitere Informationen zur Verwendung von AWS KMS mit Amazon S3, siehe [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

- b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren. S3-Bucket Keys senken die Verschlüsselungskosten, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

16. (Optional) Wenn Sie die S3-Objektsperre aktivieren möchten, gehen Sie wie folgt vor:

- a. Wählen Sie Erweiterte Einstellungen aus.

 **Important**

Durch Aktivieren der Objektsperre wird auch die Versioning für den Bucket aktiviert. Nach dem Aktivieren müssen Sie die Standardeinstellungen für die Objektsperre im

Hinblick auf die (rechtliche) Aufbewahrung konfigurieren, um neue Objekte vor dem Löschen oder Überschreiben zu schützen.

- b. Wenn Sie die Objektsperre aktivieren möchten, wählen Sie Enable (Aktivieren) aus, lesen Sie die angezeigte Warnung und bestätigen Sie sie.

Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Note

Wenn Sie einen Bucket mit aktivierter Objektsperre erstellen möchten, benötigen Sie die folgenden Berechtigungen: `s3:CreateBucket`, `s3:PutBucketVersioning` und `s3:PutBucketObjectLockConfiguration`.

17. Wählen Sie Create Bucket (Bucket erstellen) aus.

Sie haben einen Bucket in Amazon S3 erstellt.

Nächster Schritt

Informationen zum Hinzufügen eines Objekts zu einem Bucket finden Sie unter [Schritt 2: Hochladen eines Objekts in Ihren Bucket](#).

Schritt 2: Hochladen eines Objekts in Ihren Bucket

Nachdem Sie in Amazon S3 einen Bucket erstellt haben, können Sie ein Objekt hochladen. Ein Objekt kann eine beliebige Datei sein: eine Textdatei, ein Foto, ein Video usw.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Ein Objekt in einen Bucket hochladen

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie in der Liste Bucket den Namen des Buckets aus, zu dem Sie Ihr Objekt hochladen möchten.
3. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
4. Wählen Sie unter Files and folders (Dateien und Ordner) die Option Add files (Dateien hinzufügen) aus.
5. Wählen Sie eine hochzuladende Datei und dann Öffnen aus.
6. Klicken Sie auf Hochladen.

Sie haben erfolgreich ein Objekt zu Ihrem Bucket hochgeladen.

Nächster Schritt

Informationen zum Anzeigen des Objekts finden Sie unter [Schritt 3: Herunterladen eines Objekts](#).

Schritt 3: Herunterladen eines Objekts

Nachdem Sie ein Objekt in einen Bucket hochgeladen haben, können Sie Informationen über Ihr Objekt anzeigen und das Objekt auf Ihren lokalen Computer herunterladen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Verwenden der S3-Konsole

In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole ein Objekt aus einem S3-Bucket herunterladen.

Note

- Sie können jeweils nur ein Objekt herunterladen.
- Wenn Sie über die Amazon-S3-Konsole ein Objekt herunterladen, dessen Schlüsselname mit Punkt (.) endet, wird der Punkt aus dem Schlüsselnamen des heruntergeladenen

Objekts entfernt. Wenn der Punkt am Ende des Namens des heruntergeladenen Objekts beibehalten werden soll, müssen Sie die AWS Command Line Interface (AWS CLI), die AWS-SDKs oder die Amazon-S3-REST-API verwenden.

Ein Objekt von einem S3-Bucket herunterladen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, aus dem Sie ein Objekt herunterladen möchten.
3. Sie können ein Objekt wie folgt aus einem S3-Bucket herunterladen:
 - Aktivieren Sie das Kontrollkästchen neben dem Objekt und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.
 - Wenn Sie eine spezifische Version des Objekts herunterladen möchten, aktivieren Sie die Schaltfläche Versionen anzeigen (neben dem Suchfeld). Aktivieren Sie das Kontrollkästchen neben der gewünschten Version des Objekts und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.

Sie haben Ihr Objekt erfolgreich heruntergeladen.

Nächster Schritt

Informationen zum Kopieren und Einfügen Ihres Objekts in Amazon S3 finden Sie unter [Schritt 4: Kopieren Ihres Objekts in einen Ordner](#).

Schritt 4: Kopieren Ihres Objekts in einen Ordner

Sie haben einem Bucket bereits ein Objekt hinzugefügt und das Objekt heruntergeladen. Jetzt erstellen Sie einen Ordner, kopieren das Objekt und fügen es in den Ordner ein.

 Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

So kopieren Sie ein Objekt in einen Ordner

1. Wählen Sie in der Liste Buckets (Buckets) den Namen Ihres Buckets.
2. Wählen Sie Create folder (Ordner erstellen) aus und konfigurieren Sie einen neuen Ordner:
 - a. Geben Sie einen Ordernamen ein (z. B. favorite-pics).
 - b. Als Einstellung für die Ordnerschlüsselung wählen Sie Disable (Deaktivieren) aus.
 - c. Wählen Sie Speichern.
3. Navigieren Sie zum Amazon-S3-Bucket oder -Ordner, der die Objekte enthält, die Sie kopieren möchten.
4. Markieren Sie das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.
5. Wählen Sie Actions (Aktionen) und wählen Sie aus der angezeigten Liste der Optionen die Option Copy (Kopieren) aus.

Oder wählen Sie aus den Optionen oben rechts die Option Copy (Kopieren) aus.

6. Wählen Sie den Zielordner aus:
 - a. Wählen Sie S3 durchsuchen.
 - b. Wählen Sie das Optionsfeld links vom Ordernamen aus.

Um in einen Ordner zu navigieren und einen Unterordner als Ziel auszuwählen, wählen Sie den Ordernamen aus.

- c. Wählen Sie Choose destination (Ziel wählen).

Der Pfad zu Ihrem Zielordner wird im Feld Destination (Ziel) angezeigt. In Destination (Ziel) können Sie alternativ Ihren Zielpfad eingeben, z. B. `s3://bucket-name/folder-name`.

7. Wählen Sie unten rechts Copy (Kopieren).

Amazon S3 kopiert Ihre Objekte in den Zielordner.

Nächster Schritt

Informationen zum Löschen eines Objekts und eines Buckets in Amazon S3 finden Sie unter [Schritt 5: Löschen Ihrer Objekte und Buckets](#).

Schritt 5: Löschen Ihrer Objekte und Buckets

Wenn Sie ein Objekt oder einen Bucket mehr benötigen, sollten Sie diese löschen, um weitere Gebühren zu vermeiden. Wenn Sie diesen Walkthrough für die ersten Schritte zur Übung durchgeführt haben und Ihren Bucket oder Ihre Objekte nicht verwenden möchten, sollten Sie den Bucket und die Objekte löschen, damit keine weiteren Gebühren anfallen.

Bevor Sie Ihren Bucket löschen, leeren Sie den Bucket oder löschen Sie die Objekte im Bucket. Nachdem Sie Ihre Objekte und Ihren Bucket gelöscht haben, sind die Objekte und der Bucket nicht mehr verfügbar.

Wenn Sie denselben Bucket-Namen weiter verwenden möchten, sollten Sie die Objekte löschen oder den Bucket leeren, den Bucket selbst jedoch nicht löschen. Wenn Sie einen Bucket gelöscht haben, können Sie den Namen wiederverwenden. Ein anderes AWS-Konto kann jedoch einen Bucket mit demselben Namen erstellen, bevor Sie eine Chance hatten, den Namen wiederzuverwenden.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Löschen eines Objekts](#)
- [Leeren des Buckets](#)
- [Löschen des Buckets](#)

Löschen eines Objekts

Wenn Sie nur bestimmte Objekte löschen möchten, ohne alle Objekte im Bucket zu löschen, können Sie ein Objekt löschen.

1. Wählen Sie in der Liste Buckets (Buckets) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
2. Wählen Sie das zu löschende Objekt aus.
3. Wählen Sie in den Optionen oben rechts Löschen aus.
4. Geben Sie auf der Seite Objekte löschen **delete** ein, um das Löschen Ihrer Objekte zu bestätigen.
5. Wählen Sie Delete objects (Objekte löschen).

Leeren des Buckets

Wenn Sie Ihren Bucket löschen möchten, müssen Sie zuerst den Bucket leeren, um alle Objekte im Bucket zu löschen.

So leeren Sie einen Bucket

1. Wählen Sie in der Liste Buckets (Buckets) den Bucket aus, den Sie leeren möchten. Wählen Sie anschließend Empty (Leeren) aus.
2. Um zu bestätigen, dass Sie den Bucket leeren und alle in diesem enthaltenen Objekte löschen möchten, geben Sie unter Bucket leeren **permanently delete** ein.

Important

Das Leeren des Buckets kann nicht rückgängig gemacht werden. Dem Bucket während der Löschaktion hinzugefügte Objekte werden gelöscht.

3. Um den Bucket zu leeren und alle in diesem enthaltenen Objekte zu löschen, wählen Sie Empty (Leeren) aus.

Anschließend wird die Seite Empty bucket: Status (Bucket leeren: Status) geöffnet, auf der Sie eine Übersicht über fehlgeschlagene und erfolgreiche Objektlöschungen anzeigen können.

4. Um zur Bucket-Liste zurückzukehren, wählen Sie Exit (Beenden) aus.

Löschen des Buckets

Nachdem Sie den Bucket geleert oder alle Objekte aus dem Bucket gelöscht haben, können Sie den Bucket löschen.

1. Um einen Bucket zu löschen, wählen Sie in der Liste Buckets (Buckets) den Bucket aus.
2. Wählen Sie Delete (Löschen).
3. Geben Sie zum Bestätigen des Löschvorgangs unter Bucket löschen den Namen des Buckets ein.

Important

Das Löschen eines Buckets kann nicht rückgängig gemacht werden. Bucket-Namen sind eindeutig. Wenn Sie Ihren Bucket löschen, kann ein anderer AWS-Benutzer den Namen verwenden. Wenn Sie denselben Bucket-Namen weiter verwenden möchten, sollten Sie den Bucket nicht löschen. Sie sollten den Bucket stattdessen leeren und behalten.

4. Um den Bucket zu löschen, wählen Sie Delete bucket (Bucket löschen) aus.

Nächste Schritte

Im vorherigen Beispiel haben Sie gelernt, wie einige der grundlegenden Aufgaben in Amazon S3 erfolgen.

Die folgenden Themen erklären die Lernpfade, die Sie benutzen können, um sich ein vertieftes Wissen über Amazon S3 zu verschaffen, sodass Sie es in Ihren Anwendungen implementieren können.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Verstehen allgemeiner Anwendungsfälle](#)

- [Kontrollieren des Zugriffs auf Buckets und Objekte](#)
- [Verwalten und Überwachen Ihres Speichers](#)
- [Entwickeln mit Amazon S3](#)
- [Erwerben Sie neue Kenntnisse aus Tutorials](#)
- [Lernen Sie Trainings und Support kennen](#)

Verstehen allgemeiner Anwendungsfälle

Sie können Amazon S3 verwenden, um Ihren speziellen Anwendungsfall zu unterstützen. Die [AWS Solutions Library \(Bibliothek für Lösungen\)](#) und [AWS -Blog](#) bieten anwendungsfallsspezifische Informationen und Tutorials. Im Folgenden sind einige häufige Anwendungsfälle für Amazon S3 zu finden:

- Backup und Speicherung – Verwenden Sie Amazon S3 Speicherverwaltungsfunktionen, um Kosten zu verwalten, gesetzliche Anforderungen zu erfüllen, Latenz zu reduzieren und mehrere verschiedene Kopien Ihrer Daten für Compliance-Anforderungen zu speichern.
- Hosting von Anwendungen – Sie können Webanwendungen, die zuverlässig, hochgradig skalierbar und kostengünstig sind, bereitstellen, installieren und verwalten. Sie können beispielsweise Ihren Amazon-S3-Bucket so konfigurieren, dass er eine statische Website hostet. Weitere Informationen finden Sie unter [Hosten einer statischen Website mit Amazon S3](#).
- Medienhosting – Entwicklung einer hoch verfügbaren Infrastruktur, die Uploads und Downloads von Video-, Foto- oder Musikdateien hostet.
- Softwarebereitstellung – Hosting von Softwareanwendungen, die Kunden herunterladen können.

Kontrollieren des Zugriffs auf Buckets und Objekte

Amazon S3 enthält eine Vielzahl von Sicherheitsfunktionen und -tools. Eine Übersicht finden Sie unter [Zugriffsverwaltung](#).

Standardmäßig werden S3-Buckets und -Objekte als privat eingestuft. Sie haben nur Zugriff auf die S3-Ressourcen, die Sie erstellen. Sie können die folgenden Funktionen verwenden, um detaillierte Ressourcenberechtigungen zu erteilen, die Ihren speziellen Anwendungsfall unterstützen, oder um die Berechtigungen Ihrer Amazon S3-Ressourcen zu überprüfen.

- [S3 öffentlichen Zugriff blockieren](#) – Blockieren Sie den öffentlichen Zugriff auf S3-Buckets und -Objekte. Standardmäßig sind die Einstellungen für das Blockieren des öffentlichen Zugriffs auf Bucket-Ebene aktiviert.
- [AWS Identity and Access Management \(IAM-\) Identitäten](#) — Verwenden Sie IAM oder erstellen Sie IAM-Identitäten in Ihrem, AWS IAM Identity Center um den Zugriff auf Ihre Amazon S3 AWS-Konto S3-Ressourcen zu verwalten. Beispielsweise können Sie IAM mit Amazon S3 verwenden, um die Art des Zugriffs zu kontrollieren, den ein Benutzer oder eine Benutzergruppe auf einen Amazon S3 S3-Bucket hat, der Ihnen AWS-Konto gehört. Weitere Informationen zu unterschiedlichen IAM-Identitäten und bewährten Verfahren finden Sie unter [IAM-Identitäten](#) (Benutzer, Benutzergruppen und Rollen) im IAM-Benutzerhandbuch.
- [Bucket-Richtlinien](#) – Verwenden Sie die IAM-basierte Richtlinienprache, um ressourcenbasierte Berechtigungen für Ihre S3-Buckets und die darin enthaltenen Objekte zu konfigurieren.
- [Zugriffskontrolllisten \(ACLs\)](#) – Erteilen Sie autorisierten Benutzern Lese- und Schreibberechtigungen für einzelne Buckets und Objekte. Als allgemeine Regel sollten Sie S3-ressourcenbasierte Richtlinien (Bucket-Richtlinien und Zugriffspunkt-Richtlinien) oder IAM-Benutzerrichtlinien für die Zugriffssteuerung anstelle von ACLs verwenden. Richtlinien stellen eine vereinfachte und flexiblere Zugriffskontrolloption dar. Mit Bucket- und Zugriffspunktrichtlinien können Sie Regeln definieren, die allgemein für alle Anfragen an Ihre Amazon-S3-Ressourcen gelten. Weitere Informationen dazu, wann Sie ACLs anstelle von ressourcenbasierten Richtlinien oder IAM-Benutzerrichtlinien verwenden, finden Sie unter [Identity and Access Management für Amazon S3](#).
- [S3 Object Ownership](#) – Übernehmen Sie die Verantwortung für jedes Objekt in Ihrem Bucket, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. S3 Object Ownership ist eine Amazon-S3-Einstellung auf Bucket-Ebene, mit der Sie ACLs deaktivieren oder aktivieren können. Standardmäßig sind ACLs deaktiviert. Wenn ACLs deaktiviert sind, besitzt der Bucket-Eigentümer alle Objekte im Bucket und verwaltet den Datenzugriff ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.
- [IAM-Zugriffs-Analyzer für S3](#) – Bewerten und überwachen Sie Ihre S3-Bucket-Zugriffsrichtlinien, um sicherzustellen, dass die Richtlinien nur den beabsichtigten Zugriff auf Ihre S3-Ressourcen zulassen.

Verwalten und Überwachen Ihres Speichers

- [Verwalten Ihres Speichers](#) – Nachdem Sie Buckets erstellt und Objekte in Amazon S3 hochgeladen haben, können Sie Ihren Objektspeicher verwalten. Beispielsweise können Sie die S3-

Versionsverwaltung und Amazon-S3-Replikation für Notfallwiederherstellung, den S3-Lebenszyklus zur Verwaltung von Speicherkosten und die S3-Objektsperre zur Erfüllung der Compliance-Anforderungen verwenden.

- [Überwachung Ihres Speichers](#) — Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon S3 und Ihren AWS Lösungen. Sie können Speicheraktivitäten und -kosten überwachen. Wir empfehlen auch, dass Sie von allen Teilen Ihrer AWS -Lösung Überwachungsdaten sammeln, damit Sie Ausfälle, die sich über mehrere Punkte erstrecken, leichter debuggen können.
- [Analytik und Erkenntnisse](#) – Sie können Analytik und Erkenntnisse in Amazon S3 verwenden, um Ihre Speichernutzung zu verstehen, zu analysieren und zu optimieren. Verwenden Sie z. B. [Amazon S3 Storage Lens](#), um Ihren Speicher zu verstehen, zu analysieren und zu optimieren. S3 Storage Lens bietet mehr als 29 Nutzungs- und Aktivitätsmetriken und interaktive Dashboards, um Daten für Ihre gesamte Organisation, bestimmte Konten, Regionen, Buckets oder Präfixe zu aggregieren. Verwenden Sie [Speicherklassenanalysen](#), um Speicherzugriffsmuster zu analysieren und zu entscheiden, wann es an der Zeit ist, Ihre Daten in eine kostengünstigere Speicherklasse zu verschieben.

Entwickeln mit Amazon S3

Amazon S3 ist ein REST-Service. Sie können Anfragen über die REST-API oder die AWS SDK-Bibliotheken an Amazon S3 senden, die die zugrunde liegende Amazon S3 S3-REST-API umschließen und so Ihre Programmieraufgaben vereinfachen. Sie können auch die AWS Command Line Interface (AWS CLI) verwenden, um Amazon S3-API-Aufrufe durchzuführen. Weitere Informationen finden Sie unter [Senden von Anforderungen](#).

Die Amazon S3-REST-API ist eine HTTP-Schnittstelle zu Amazon S3. Mit REST API verwenden Sie HTTP-Standardanfragen, um Buckets und Objekte zu erstellen, laden oder löschen. Sie können einen beliebigen Toolkit einsetzen, der HTTP unterstützt, um die REST-API verwenden zu können. Sie können sogar einen Browser verwenden, um Objekte zu laden, wenn diese anonym lesbar sind. Weitere Informationen finden Sie unter [Entwickeln mit Amazon S3 unter Verwendung der REST-API](#).

Damit Sie Anwendungen in der von Ihnen bevorzugten Sprache entwickeln können, stellen wir die folgenden Ressourcen zur Verfügung.

AWS CLI

Sie können auf die Funktionen von Amazon S3 mit AWS CLI zugreifen. Informationen zum Herunterladen und Konfigurieren von finden Sie AWS CLI unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

[Das AWS CLI bietet zwei Befehlsebenen für den Zugriff auf Amazon S3: Befehle auf hoher Ebene \(s3\) und Befehle auf API-Ebene \(s3api und s3control\)](#). Die High-Level-S3-Befehle vereinfachen das Ausführen häufiger Vorgänge. Zu diesen zählen das Erstellen, Bearbeiten und Löschen von Objekten und Buckets. Die Befehle s3api und s3control bieten direkten Zugriff auf alle Amazon S3-API-Vorgänge, mit denen Sie erweiterte Vorgänge ausführen können, die möglicherweise nicht möglich sind, wenn Sie ausschließlich High-Level-Befehle verwenden.

[Eine Liste der Amazon S3 AWS CLI S3-Befehle finden Sie unter s3, s3api und s3control.](#)

AWS SDKs und Explorer

Sie können die AWS SDKs bei der Entwicklung von Anwendungen mit Amazon S3 verwenden. Die AWS -SDKs vereinfachen Ihre Programmierungsaufgaben, indem sie die zugrunde liegende REST-API umhüllen. Die AWS Mobile SDKs und die JavaScript Amplify-Bibliothek sind auch für die Erstellung verbundener Mobil- und Webanwendungen verfügbar. AWS

Zusätzlich zu den AWS SDKs sind AWS Explorers für Visual Studio und Eclipse für Java IDE verfügbar. In diesem Fall sind die SDKs und die Explorer als Toolkits gebündelt. AWS

Weitere Informationen finden Sie unter [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Beispiel-Code und Bibliotheken

Das [AWS -Entwicklerzentrum](#) und der [AWS Code-Beispiel-Katalog](#) und Bibliotheken bieten Beispiel-Code und Bibliotheken speziell für Amazon S3. Sie können mithilfe dieser Codebeispiele die Implementierung der Amazon S3-API verstehen. Sie können auch die [API-Referenz zu Amazon Simple Storage Service](#) anzeigen, um die Amazon S3-API-Vorgänge im Detail zu verstehen.

Erwerben Sie neue Kenntnisse aus Tutorials

Sie können mit step-by-step Tutorials beginnen, um mehr über Amazon S3 zu erfahren. Diese Tutorials sind für Laborumgebungen vorgesehen und sie benutzen fiktive Unternehmensnamen, Benutzernamen usw. In den Tutorials finden Sie allgemeine Anleitungen. Ohne die sorgfältige Prüfung und Anpassung an die besonderen Gegebenheiten der Umgebung Ihrer Organisation sind sie nicht zur direkten Verwendung in einer Produktionsumgebung bestimmt.

Erste Schritte

- [Tutorial: Speichern und Abrufen einer Datei mit Amazon S3](#)
- [Tutorial: Erste Schritte mit S3 Intelligent-Tiering](#)
- [Tutorial: Erste Schritte mit den Amazon-S3-Glacier-Speicherklassen](#)

Optimierung der Speicherkosten

- [Tutorial: Erste Schritte mit S3 Intelligent-Tiering](#)
- [Tutorial: Erste Schritte mit den Amazon-S3-Glacier-Speicherklassen](#)
- [Tutorial: Optimieren von Kosten und Transparenz bei der Nutzung mit S3 Storage Lens](#)

Verwalten des Speichers

- [Tutorial: Erste Schritte mit Multi-Region-Zugangspunkten in Amazon S3](#)
- [Tutorial: Replizieren vorhandener Objekte in Ihren Amazon-S3-Buckets mit S3 Batch Replication](#)

Hosten von Videos und Websites

- [Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53](#)
- [Tutorial: Konfigurieren einer statischen Website auf Amazon S3](#)
- [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#)

Verarbeiten von Daten

- [Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#)
- [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#)
- [Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen](#)
- [Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations, AWS Lambda, und AWS Elemental MediaConvert](#)

Schutz von Daten

- [Tutorial: Überprüfen der Integrität von Daten in Amazon S3 mit zusätzlichen Prüfsummen](#)
- [Tutorial: Replizieren von Daten innerhalb und zwischen Daten AWS-Regionen mithilfe von S3 Replication](#)
- [Tutorial: Schutz von Daten in Amazon S3 vor dem versehentlichen Löschen oder Anwendungsfehlern mithilfe von S3 Versioning, S3 Object Lock und S3 Replication](#)
- [Tutorial: Replizieren vorhandener Objekte in Ihren Amazon S3-Buckets mit S3 Batch Replication](#)

Lernen Sie Trainings und Support kennen

Sie können von AWS Experten lernen, um Ihre Fähigkeiten zu erweitern und Expertenunterstützung bei der Erreichung Ihrer Ziele zu erhalten.

- Training – Trainingsressourcen bieten einen praktischen Ansatz zum Erlernen von Amazon S3. Weitere Informationen finden Sie unter [AWS -Schulungen und -Zertifizierung](#) und [AWS -Online-Kundendienstmitarbeiter](#).
- Diskussionsforen – Im Forum können Sie Beiträge überprüfen, um zu erfahren, was mit Amazon S3 alles möglich ist und was nicht. Sie können auch Ihre Fragen posten. Weitere Informationen finden Sie unter [Diskussionsforen](#).
- Technischer Support – Wenn Sie weitere Fragen haben, können Sie sich an den [Technischen Support](#) wenden.

Tutorials

In den folgenden Tutorials end-to-end werden vollständige Verfahren für allgemeine Amazon S3 S3-Aufgaben vorgestellt. Diese Tutorials sind für Laborumgebungen vorgesehen und sie benutzen fiktive Unternehmensnamen, Benutzernamen usw. In den Tutorials finden Sie allgemeine Anleitungen. Ohne die sorgfältige Prüfung und Anpassung an die besonderen Gegebenheiten der Umgebung Ihrer Organisation sind sie nicht zur direkten Verwendung in einer Produktionsumgebung bestimmt.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Erste Schritte

- [Tutorial: Speichern und Abrufen einer Datei mit Amazon S3](#)
- [Tutorial: Erste Schritte mit S3 Intelligent-Tiering](#)
- [Tutorial: Erste Schritte mit den Amazon-S3-Glacier-Speicherklassen](#)

Optimierung der Speicherkosten

- [Tutorial: Erste Schritte mit S3 Intelligent-Tiering](#)
- [Tutorial: Erste Schritte mit den Amazon-S3-Glacier-Speicherklassen](#)
- [Tutorial: Optimieren von Kosten und Transparenz bei der Nutzung mit S3 Storage Lens](#)

Verwalten des Speichers

- [Tutorial: Erste Schritte mit Multi-Region-Zugangspunkten in Amazon S3](#)
- [Tutorial: Replizieren vorhandener Objekte in Ihren Amazon-S3-Buckets mit S3 Batch Replication](#)

Hosten von Videos und Websites

- [Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53](#)
- [Tutorial: Konfigurieren einer statischen Website auf Amazon S3](#)
- [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#)

Verarbeiten von Daten

- [Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#)
- [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#)
- [Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen](#)
- [Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations, AWS Lambda, und AWS Elemental MediaConvert](#)

Schutz von Daten

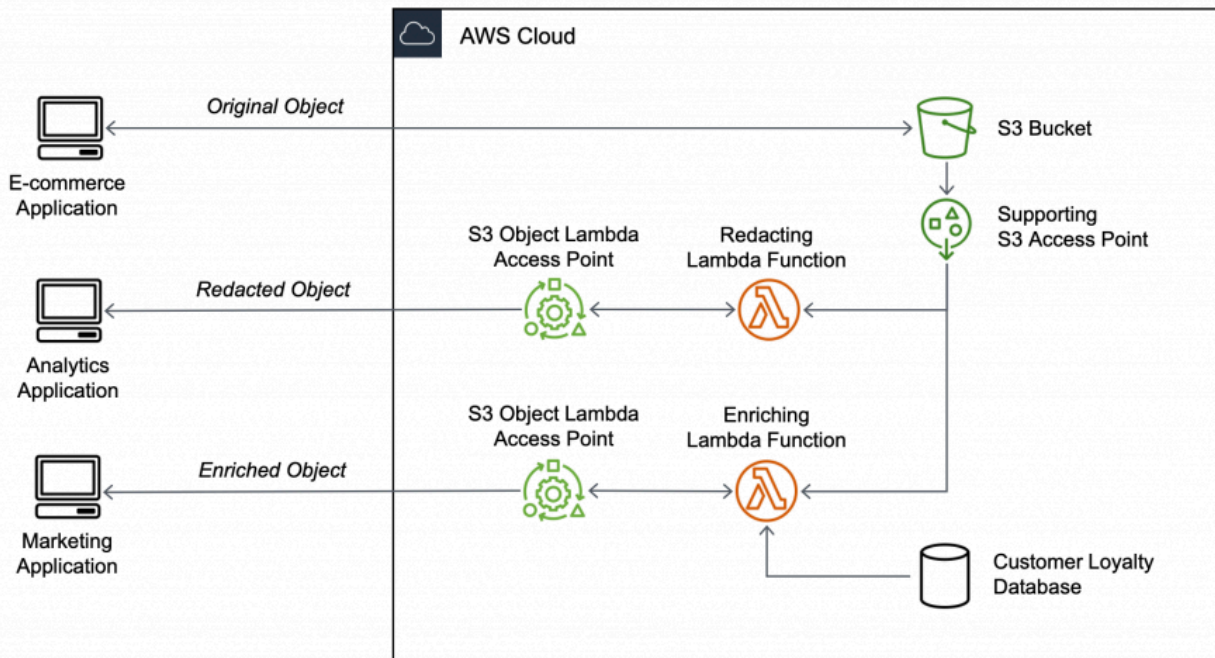
- [Tutorial: Überprüfen der Integrität von Daten in Amazon S3 mit zusätzlichen Prüfsummen](#)
- [Tutorial: Daten innerhalb und zwischen Daten replizieren AWS-Regionen mit S3-Replikation](#)
- [Tutorial: Schutz von Daten in Amazon S3 vor dem versehentlichen Löschen oder Anwendungsfehlern mithilfe von S3 Versioning, S3 Object Lock und S3 Replication](#)
- [Tutorial: Replizieren vorhandener Objekte in Ihren Amazon-S3-Buckets mit S3 Batch Replication](#)

Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda

Wenn Sie Daten in Amazon S3 speichern, können Sie sie für mehrere Anwendungen verwenden. Jede Anwendung hat jedoch möglicherweise eindeutige Anforderungen an das Datenformat und benötigt möglicherweise eine Änderung oder Verarbeitung Ihrer Daten für einen bestimmten Anwendungsfall. Beispielsweise kann ein von einer E-Commerce-Anwendung erstellter Datensatz personenbezogene Daten (PII) enthalten. Wenn dieselben Daten für Analysen verarbeitet werden,

PII ist dies nicht erforderlich und sollte redigiert werden. Wenn jedoch derselbe Datensatz für eine Marketingkampagne verwendet wird, müssen Sie die Daten möglicherweise mit zusätzlichen Details wie Informationen aus der Kundenbindungsdatenbank anreichern.

Mit [S3 Object Lambda](#) können Sie Ihren eigenen Code hinzufügen, um Daten zu verarbeiten, die von S3 abgerufen werden, bevor Sie sie an eine Anwendung zurückgeben. Insbesondere können Sie eine konfigurieren AWS Lambda Funktion und hängen Sie sie an einen S3 Object Lambda Access Point an. Wenn eine Anwendung [standardmäßige GET S3-Anfragen über den S3](#) Object Lambda Access Point sendet, wird die angegebene Lambda-Funktion aufgerufen, um alle Daten zu verarbeiten, die aus einem S3-Bucket über den unterstützenden S3-Zugriffspunkt abgerufen wurden. Dann gibt der S3 Object Lambda Access Point das transformierte Ergebnis zurück an die Anwendung aus. Sie können Ihre eigenen benutzerdefinierten Lambda-Funktionen erstellen und ausführen, indem Sie die S3 Object Lambda-Datentransformation an Ihren spezifischen Anwendungsfall anpassen, ohne dass Änderungen an Ihren Anwendungen erforderlich sind.



Ziel

In diesem Tutorial erfahren Sie, wie Sie standardmäßigen GET S3-Anfragen benutzerdefinierten Code hinzufügen, um das von S3 abgerufene angeforderte Objekt so zu ändern, dass das Objekt den Anforderungen des anfragenden Clients oder der anfragenden Anwendung entspricht. Insbesondere erfahren Sie, wie Sie den gesamten Text im ursprünglichen Objekt, das in S3 gespeichert ist, über S3 Object Lambda in Großbuchstaben umwandeln.

Note

Dieses Tutorial verwendet Python-Code, um die Daten zu transformieren, zum Beispiel mit anderen AWS SDKs siehe [Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#) im AWS SDK Bibliothek mit Codebeispielen.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Einen S3-Bucket erstellen](#)
- [Schritt 2: Hochladen einer Datei zu einem S3-Bucket](#)
- [Schritt 3: Erstellen eines S3-Zugriffspunkts](#)
- [Schritt 4: Erstellen einer Lambda-Funktion](#)
- [Schritt 5: Konfigurieren Sie eine IAM Richtlinie für die Ausführungsrolle Ihrer Lambda-Funktion](#)
- [Schritt 6: Erstellen eines S3 Object Lambda Access Point](#)
- [Schritt 7: Anzeigen der transformierten Daten](#)
- [Schritt 8: Bereinigen](#)
- [Nächste Schritte](#)

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie eine AWS-Konto bei dem Sie sich anmelden können als AWS Identity and Access Management (IAM) Benutzer mit den richtigen Berechtigungen. Sie müssen außerdem Python Version 3.8 oder höher installieren.

Teilschritte

- [Erstellen Sie einen IAM Benutzer mit Berechtigungen in Ihrem AWS-Konto \(Konsole\)](#)
- [Installieren Sie Python 3.8 oder höher auf Ihrem lokalen Computer](#)

Erstellen Sie einen IAM Benutzer mit Berechtigungen in Ihrem AWS-Konto (Konsole)

Sie können einen IAM Benutzer für das Tutorial erstellen. Um dieses Tutorial abzuschließen, muss Ihr IAM Benutzer die folgenden IAM Richtlinien für den Zugriff auf relevante Dokumente beifügen

AWS Ressourcen und bestimmte Aktionen ausführen. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie im Benutzerhandbuch unter [IAMBenutzer erstellen \(Konsole\)](#). IAM

Ihr IAM Benutzer benötigt die folgenden Richtlinien:

- [AmazonS3 FullAccess — Erteilt](#) Berechtigungen für alle Amazon S3-Aktionen, einschließlich der Berechtigungen zum Erstellen und Verwenden eines Object Lambda Access Points.
- [AWSLambda_FullAccess](#) — Erteilt Berechtigungen für alle Lambda-Aktionen.
- [IAMFullAccess](#)— Erteilt Berechtigungen für alle IAM Aktionen.
- [IAMAccessAnalyzerReadOnlyAccess](#)— Erteilt Berechtigungen zum Lesen aller von Access Analyzer bereitgestellten IAM Zugriffsinformationen.
- [CloudWatchLogsFullAccess](#)— Gewährt vollen Zugriff auf CloudWatch Logs.

Note

Der Einfachheit halber erstellt und verwendet dieses Tutorial einen IAM Benutzer. Denken Sie nach Abschluss dieses Tutorials an [Löschen Sie den IAM Benutzer](#). Für den produktiven Einsatz empfehlen wir, dass Sie die [bewährten Sicherheitsmethoden IAM im IAM](#) Benutzerhandbuch befolgen. Eine bewährte Methode sieht vor, dass menschliche Benutzer für den Zugriff einen Verbund mit einem Identitätsanbieter verwenden AWS mit temporären Anmeldeinformationen. Eine weitere bewährte Methode besteht darin, zu verlangen, dass Workloads temporäre Anmeldeinformationen mit IAM Rollen für den Zugriff verwenden AWS. Um mehr über die Verwendung zu erfahren AWS IAM Identity Center Informationen zum Erstellen von Benutzern mit temporären Anmeldeinformationen finden Sie unter [Erste Schritte](#) im AWS IAM Identity Center Benutzerleitfaden.

In diesem Tutorial wird auch Vollzugriff verwendet AWS verwaltete Richtlinien. Für die Verwendung in der Produktion empfehlen wir, dass Sie stattdessen nur die für Ihren Anwendungsfall erforderlichen Mindestberechtigungen gemäß [Bewährte Methoden in Bezug auf die Sicherheit](#) erteilen.

Installieren Sie Python 3.8 oder höher auf Ihrem lokalen Computer

Verwenden Sie das folgende Verfahren, um Python 3.8 oder höher auf Ihrem lokalen Computer zu installieren. Weitere Installationsanweisungen finden Sie auf der Seite [Python herunterladen](#) im Python-Anfängerhandbuch.

1. Öffnen Sie das lokale Terminal oder die Shell, und führen Sie den folgenden Befehl aus, um festzustellen, ob Python bereits installiert ist, und wenn ja, welche Version installiert ist.

```
python --version
```

2. Wenn Sie nicht über Python 3.8 oder höher verfügen, laden Sie die Datei [offizieller Installer](#) von Python 3.8 oder höher herunter, die für Ihren lokalen Computer geeignet ist.
3. Führen Sie das Installationsprogramm aus, indem Sie auf die heruntergeladene Datei doppelklicken und die Schritte ausführen, um die Installation abzuschließen.

Für Windows-Benutzer wählen Sie Python 3.X hinzufügen PATH im Installationsassistenten, bevor Sie Jetzt installieren wählen.

4. Starten Sie das Terminal neu, indem Sie es schließen und erneut öffnen.
5. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Python 3.8 oder höher ordnungsgemäß installiert ist.

Führen Sie für macOS-Benutzer diesen Befehl aus:

```
python3 --version
```

Führen Sie für Windows-Benutzer diesen Befehl aus:

```
python --version
```

6. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der pip3-Paket-Manager installiert ist. Wenn Sie in der Befehlsantwort eine Pip-Versionsnummer und Python 3.8 oder höher sehen, bedeutet dies, dass der pip3-Paketmanager erfolgreich installiert wurde.

```
pip --version
```

Schritt 1: Einen S3-Bucket erstellen

Erstellen Sie einen Bucket zum Speichern der ursprünglichen Daten, die Sie transformieren möchten.

So erstellen Sie einen Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

Die Seite Create bucket (Bucket erstellen) wird geöffnet.

4. Geben Sie für Bucket-Name einen Namen für Ihren Bucket ein (z. B. **tutorial-bucket**).

Weitere Informationen zur Benennung von Buckets in Amazon S3 finden Sie unter [Regeln für die Benennung von Buckets](#).

5. Wählen Sie für Region die AWS-Region wo sich der Bucket befinden soll.

Weitere Informationen zur Bucket-Region finden Sie unter [Bucket-Übersicht](#).

6. Belassen Sie die Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren bei den Standardeinstellungen (Alle öffentlichen Zugriffe blockieren ist aktiviert).

Es wird empfohlen, alle Einstellungen für öffentlichen Zugriff blockieren aktiviert zu belassen, es sei denn, Sie müssen eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

(Optional) Informationen zum Konfigurieren zusätzlicher Bucket-Einstellungen für Ihren spezifischen Anwendungsfall finden Sie unter [Erstellen eines Buckets](#).

8. Wählen Sie Bucket erstellen aus.

Schritt 2: Hochladen einer Datei zu einem S3-Bucket

Laden Sie die Textdatei zu einem S3-Bucket hoch. Diese Textdatei enthält die Originaldaten, die Sie später in diesem Tutorial in Großbuchstaben umwandeln.

Sie können beispielsweise eine `tutorial.txt`-Datei, die den folgenden Text enthält, hochladen:

```
Amazon S3 Object Lambda Tutorial:  
You can add your own code to process data retrieved from S3 before  
returning it to an application.
```


Hochladen einer Datei in einen Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben, in den Sie die Datei hochladen möchten.
4. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
5. Wählen Sie auf der Seite Upload unter Dateien und Ordner die Option Dateien hinzufügen aus.
6. Wählen Sie eine hochzuladende Datei und dann Open (Öffnen) aus. Sie können z. B. das oben erwähnte `tutorial.txt`-Dateibeispiel hochladen.
7. Klicken Sie auf Hochladen.

Schritt 3: Erstellen eines S3-Zugriffspunkts

Um einen S3 Object Lambda Access Point für den Zugriff und die Umwandlung der ursprünglichen Daten zu verwenden, müssen Sie einen S3-Zugriffspunkt erstellen und dem S3-Bucket zuordnen, den Sie in [Schritt 1](#) erstellt haben. Der Access Point muss sich im selben befinden AWS-Region wie die Objekte, die Sie transformieren möchten.

Später in diesem Tutorial verwenden Sie diesen Zugriffspunkt als unterstützenden Zugriffspunkt für Ihren Object Lambda Access Point.

So erstellen Sie einen Zugangspunkt

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Wählen Sie auf der Seite Zugriffspunkte die Option Zugriffspunkt erstellen aus.
4. Geben Sie im Feld Name des Zugriffspunkts den gewünschten Namen für den Zugriffspunkt ein (z. B. **tutorial-access-point**).

Weitere Hinweise zur Benennung von Zugangspunkten finden Sie unter [Regeln zur Benennung von Amazon S3-Zugriffspunkten](#).

5. Wählen Sie im Feld Bucket-Name den Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben. S3 hängt den Zugriffspunkt an diesen Bucket an.

(Optional) Sie können S3 durchsuchen auswählen, um die Buckets in Ihrem Konto zu browsen und zu durchsuchen. Wenn Sie Browse S3 (S3 durchsuchen) wählen, wählen Sie den gewünschten Bucket aus und dann Choose path (Pfad wählen), um das Feld Bucket name (Bucket-Name) mit dem Namen dieses Buckets zu füllen.

6. Wählen Sie für Netzwerkursprung Internetaus.

Weitere Informationen zu Netzwerkursprüngen für Zugangspunkte finden Sie unter [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#).

7. Alle Block-Public-Access-Einstellungen sind standardmäßig für Zugriffspunkte aktiviert. Sie sollten alle Einstellungen für das Blockieren aller öffentlichen Zugriffe aktiviert lassen.

Weitere Informationen finden Sie unter [Verwalten des öffentlichen Zugriffs auf Zugriffspunkte](#).

8. Behalten Sie für alle anderen Zugriffspunkteinstellungen die Standardeinstellungen bei.

(Optional) Sie können die Zugriffspunkteinstellungen ändern, um Ihren Anwendungsfall zu unterstützen. Für dieses Tutorial empfehlen wir, dass die Standardeinstellungen beibehalten werden.

(Optional) Wenn Sie den Zugriff auf Ihren Zugriffspunkt verwalten müssen, können Sie eine Zugriffspunkt-Richtlinie angeben. Weitere Informationen finden Sie unter [Beispiele von - Zugriffspunktrichtlinien](#).

9. Wählen Sie Create access point (Zugriffspunkt erstellen) aus.

Schritt 4: Erstellen einer Lambda-Funktion

Erstellen Sie zum Transformieren der ursprünglichen Daten eine Lambda-Funktion zur Verwendung mit dem S3 Object Lambda Access Point.

Teilschritte

- [Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets mit einer virtuellen Umgebung](#)
- [Erstellen einer Lambda-Funktion mit einer Ausführungsrolle \(Konsole\)](#)
- [Stellen Sie Ihren Lambda-Funktion-Code mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion \(Konsole\)](#)

Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets mit einer virtuellen Umgebung

1. Erstellen Sie auf dem lokalen Computer einen Ordner mit dem Ordernamen `object-lambda` für die virtuelle Umgebung, die Sie zu einem späteren Zeitpunkt in diesem Tutorial verwenden können.
2. Erstellen Sie im `object-lambda`-Ordner eine Datei mit einer Lambda-Funktion, die den gesamten Text im ursprünglichen Objekt in Großbuchstaben ändert. Sie können z. B. die folgende Funktion verwenden, die in Python geschrieben wurde. Speichern Sie diese Funktion in einer Datei mit dem Namen `transform.py`.

```
import boto3
import requests
from botocore.config import Config

# This function capitalizes all text in the original object
def lambda_handler(event, context):
    object_context = event["getObjectContext"]
    # Get the presigned URL to fetch the requested original object
    # from S3
    s3_url = object_context["inputS3Url"]
    # Extract the route and request token from the input context
    request_route = object_context["outputRoute"]
    request_token = object_context["outputToken"]

    # Get the original S3 object using the presigned URL
    response = requests.get(s3_url)
    original_object = response.content.decode("utf-8")

    # Transform all text in the original object to uppercase
    # You can replace it with your custom code based on your use case
    transformed_object = original_object.upper()

    # Write object back to S3 Object Lambda
    s3 = boto3.client('s3', config=Config(signature_version='s3v4'))
    # The WriteGetObjectResponse API sends the transformed data
    # back to S3 Object Lambda and then to the user
    s3.write_get_object_response(
        Body=transformed_object,
        RequestRoute=request_route,
        RequestToken=request_token)
```

```
# Exit the Lambda function: return the status code
return {'status_code': 200}
```

Note

Die vorangehende Lambda-Beispielfunktion lädt das gesamte angeforderte Objekt in den Speicher, bevor es transformiert und an den Client zurückgegeben wird. Alternativ können Sie das Objekt aus S3 streamen, um das Laden des gesamten Objekts in den Speicher zu vermeiden. Diese Methode ist nützlich, wenn Sie mit großen Objekten arbeiten. Weitere Informationen zum Streamen von Antworten mit Object Lambda Access Points finden Sie in den Streaming-Beispielen in [Arbeiten mit GetObject-Anforderungen in Lambda](#).

Wenn Sie eine Lambda-Funktion für die Verwendung mit einem S3 Object Lambda Access Point schreiben, basiert die Funktion auf dem Eingabe-Ereigniskontext, den S3 Object Lambda für die Lambda-Funktion bereitstellt. Der Ereigniskontext stellt Informationen über die Anforderung bereit, die in dem von S3 Object Lambda an Lambda übergebenen Ereignis gestellt wird. Sie enthält die Parameter, die Sie verwenden, um die Lambda-Funktion zu erstellen.

Die Felder, die zum Erstellen der vorhergehenden Lambda-Funktion verwendet werden, lauten wie folgt:

Das Feld von `getObjectContext` bezeichnet die Eingabe- und Ausgabedetails für Verbindungen zu Amazon S3 und S3 Object Lambda. Es enthält die folgenden Felder:

- `inputS3Url`— Ein vorsigniertes URL, mit dem die Lambda-Funktion das Originalobjekt vom unterstützenden Access Point herunterladen kann. Durch die Verwendung eines vorsignierten URL Objekts benötigt die Lambda-Funktion keine Amazon S3 S3-Leseberechtigungen, um das ursprüngliche Objekt abzurufen, und sie kann nur auf das Objekt zugreifen, das bei jedem Aufruf verarbeitet wurde.
- `outputRoute`— Ein Routing-Token, das dem S3-Objekt-Lambda hinzugefügt wird, URL wenn die Lambda-Funktion aufruft `writeGetObjectResponse`, um das transformierte Objekt zurückzusenden.

- `outputToken` – Ein Token, das von S3 Object Lambda verwendet wird, um den `WriteGetObjectResponse`-Aufruf mit dem ursprünglichen Anrufer abzugleichen, wenn das transformierte Objekt zurücksendet wird.

Weitere Informationen zu allen Feldern im Ereigniskontext finden Sie unter [Format und Verwendung des Ereigniskontexts](#) und [Schreiben von Lambda-Funktionen für S3 Object Lambda Access Points](#).

3. Geben Sie in Ihrem lokalen Terminal den folgenden Befehl ein, um das `virtualenv`-Paket zu installieren:

```
python -m pip install virtualenv
```

4. Öffnen Sie in Ihrem lokalen Terminal den `object-lambda`-Ordner, den Sie zuvor erstellt haben und geben Sie dann den folgenden Befehl ein, um eine virtuelle Umgebung namens `venv` zu erstellen und zu initialisieren.

```
python -m virtualenv venv
```

5. Geben Sie zum Aktivieren der virtuellen Umgebung den folgenden Befehl ein, um die `activate`-Datei aus dem Ordner der Umgebung auszuführen:

Führen Sie für macOS-Benutzer diesen Befehl aus:

```
source venv/bin/activate
```

Führen Sie für Windows-Benutzer diesen Befehl aus:

```
.\venv\Scripts\activate
```

Jetzt ändert sich die Eingabeaufforderung und zeigt `(venv)` an, was darauf hinweist, dass die virtuelle Umgebung aktiv ist.

6. Um die erforderlichen Bibliotheken zu installieren, führen Sie die folgenden Befehle Zeile für Zeile in der virtuellen `venv`-Umgebung aus.

Mit diesen Befehlen werden aktualisierte Versionen der Abhängigkeiten Ihrer `lambda_handler`-Lambda -Funktion installiert. Diese Abhängigkeiten sind AWS SDK für Python (Boto3) und das Request-Modul.

```
pip3 install boto3
```

```
pip3 install requests
```

7. Sie können die virtuelle Umgebung deaktivieren, indem Sie den folgenden Befehl ausführen:

```
deactivate
```

8. Um ein Bereitstellungspaket mit den installierten Bibliotheken als `.zip`-Datei mit dem Namen `lambda.zip` an der Wurzel des `object-lambda`-Verzeichnisses zu erstellen, führen Sie die folgenden Befehle Zeile für Zeile in Ihrem lokalen Terminal aus.

 Tip

Die folgenden Befehle müssen möglicherweise angepasst werden, um in Ihrer bestimmten Umgebung zu funktionieren. Eine Bibliothek kann beispielsweise in `site-packages` oder `dist-packages` erscheinen, und der erste Ordner könnte `lib` oder `lib64` sein. Außerdem kann der Ordner `python` mit einer anderen Python-Version benannt werden. Sie können den Befehl `pip show` verwenden, um ein spezielles Paket zu finden.

Führen Sie für macOS-Benutzer diese Befehle aus:

```
cd venv/lib/python3.8/site-packages
```

```
zip -r ../../../../lambda.zip .
```

Führen Sie für Windows-Benutzer diese Befehle aus:

```
cd .\venv\Lib\site-packages\
```

```
powershell Compress-Archive * ../../../../lambda.zip
```

Der letzte Befehl speichert das Bereitstellungspaket im Stamm des `object-lambda`-Verzeichnisses.

9. Fügen Sie dem Stamm Ihres Bereitstellungspakets die Funktionscode-Datei `transform.py` hinzu.

Führen Sie für macOS-Benutzer diese Befehle aus:

```
cd ../../../../
```

```
zip -g lambda.zip transform.py
```

Führen Sie für Windows-Benutzer diese Befehle aus:

```
cd ..\..\..\
```

```
powershell Compress-Archive -update transform.py lambda.zip
```

Nach diesem Schritt sollte Ihre Verzeichnisstruktur wie folgt aussehen:

```
lambda.zip$  
# transform.py  
# __pycache__  
| boto3/  
# certifi/  
# pip/  
# requests/  
...
```

Erstellen einer Lambda-Funktion mit einer Ausführungsrolle (Konsole)

1. Melden Sie sich an bei AWS Management Console und öffne das AWS Lambda Konsole bei <https://console.aws.amazon.com/lambda/>.
2. Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
3. Wählen Sie Create function (Funktion erstellen).
4. Wählen Sie Author from scratch aus.
5. Führen Sie unter Basic information (Grundlegende Informationen) die folgenden Schritte aus:

- a. Geben Sie für Function name (Funktionsname) **tutorial-object-lambda-function** ein.
 - b. Wählen Sie für Laufzeit Python 3.8 oder eine höhere Version aus.
6. Erweitern Sie den Abschnitt Ändern der standardmäßigen Ausführungsrolle. Wählen Sie in Execution role (Ausführungsrolle) die Option Create a new role with basic Lambda permissions (Neue Rolle mit Lambda-Basisberechtigungen erstellen) aus.

In [Schritt 5](#) weiter unten in diesem Tutorial fügen Sie AmazonS3 der Ausführungsrolle ObjectLambdaExecutionRolePolicy dieser Lambda-Funktion hinzu.

7. Lassen Sie die restlichen Einstellungen bei den Standardwerten.
8. Wählen Sie Funktion erstellen aus.

Stellen Sie Ihren Lambda-Funktion-Code mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion (Konsole)

1. Im AWS Lambda Konsole unter <https://console.aws.amazon.com/lambda/>, wählen Sie im linken Navigationsbereich Funktionen aus.
2. Wählen Sie die Lambda-Funktion aus, die Sie zuvor erstellt haben (z. B **tutorial-object-lambda-function**).
3. Wählen Sie auf der Seite mit den Details der Lambda-Funktion den Code-Tab. Wählen Sie im Abschnitt Quellcode die Option Upload von und dann ZIP-Datei aus.
4. Wählen Sie Upload (Hochladen) aus, um Ihre lokale .zip-Datei auszuwählen.
5. Wählen Sie die zuvor erstellte lambda.zip-Datei und klicken Sie dann auf Öffnen.
6. Wählen Sie Save (Speichern) aus.
7. Wählen Sie im Abschnitt Laufzeiteinstellungen die Option Bearbeiten aus.
8. Bestätigen Sie auf der Seite Bearbeiten von Laufzeiteinstellungen, dass Laufzeit auf Python 3.8 oder eine höhere Version gesetzt wird.
9. Um der Lambda-Laufzeit mitzuteilen, welche Handler-Methode in Ihrem Lambda-Funktionscode aufgerufen werden soll, geben Sie **transform.lambda_handler** für Handler ein.

Wenn Sie eine Funktion in Python konfigurieren, besteht der Wert der Handler-Einstellung aus dem Dateinamen und dem Namen des Handler-Moduls, getrennt durch einen Punkt. Beispielsweise ruft `transform.lambda_handler` die `lambda_handler`-Methode auf, die in der `transform.py`-Datei definiert ist.

10. Wählen Sie Save (Speichern) aus.
11. (Optional) Wählen Sie auf der Detailseite der Lambda-Funktion den Konfiguration-Tab. Wählen Sie im linken Navigationsbereich die Option Allgemeine Konfiguration und anschließend Bearbeiten aus. Geben Sie im Timeout-Feld **1 m 0 s** ein. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt und wählen Sie Speichern aus.

Timeout ist die Zeitspanne, die Lambda einer Funktion für einen Aufruf zulässt, bevor diese gestoppt wird. Der Standardwert ist 3 Sekunden. Die maximale Dauer für eine Lambda-Funktion, die von S3 Object Lambda verwendet wird, beträgt 60 Sekunden. Die Preise basieren auf dem konfigurierten Arbeitsspeicher und der Zeit, für die der Code ausgeführt wird.

Schritt 5: Konfigurieren Sie eine IAM Richtlinie für die Ausführungsrolle Ihrer Lambda-Funktion

Damit Ihre Lambda-Funktion dem GetObject Aufrufer benutzerdefinierte Daten und Answerheader bereitstellen kann, muss die Ausführungsrolle Ihrer Lambda-Funktion über IAM Berechtigungen zum Aufrufen von verfügen. WriteGetObjectResponse API

So fügen Sie Ihrer Lambda-Funktionsrolle eine IAM Richtlinie hinzu

1. Im AWS Lambda Wählen Sie in der Konsole unter <https://console.aws.amazon.com/lambda/> Funktionen im linken Navigationsbereich aus.
2. Wählen Sie die Funktion aus, die Sie in [Schritt 4](#) erstellt haben (z. B. **tutorial-object-lambda-function**).
3. Wählen Sie auf der Detailseite Ihrer Lambda-Funktion den Tag Konfiguration und danach Berechtigungen im linken Navigationsbereich.
4. Wählen Sie unter Ausführungsrolle im Bereich den Link des Rollennamens aus. Die IAM Konsole wird geöffnet.
5. Wählen Sie auf der Übersichtsseite der IAM Konsole für die Ausführungsrolle Ihrer Lambda-Funktion den Tab Berechtigungen aus. Wählen Sie dann im Menü Add Permissions (Berechtigungen hinzufügen) die Option Attach policies (Richtlinien anfügen) aus.
6. Geben Sie auf der Seite Attach permissions (Berechtigungen anfügen) **AmazonS3ObjectLambdaExecutionRolePolicy** in das Suchfeld ein, um die Liste der Richtlinien zu filtern. Aktivieren Sie das Kontrollkästchen neben dem Namen der ObjectLambdaExecutionRolePolicyAmazonS3-Richtlinie.

7. Wählen Sie Richtlinien anfügen.

Schritt 6: Erstellen eines S3 Object Lambda Access Point

Ein S3 Object Lambda Access Point bietet die Flexibilität, eine Lambda-Funktion direkt von einer GET S3-Anfrage aus aufzurufen, sodass die Funktion Daten verarbeiten kann, die von einem S3-Zugangspunkt abgerufen wurden. Beim Erstellen und Konfigurieren eines S3 Object Lambda Access Points müssen Sie die Lambda-Funktion angeben, die aufgerufen werden soll, und den Ereigniskontext im JSON Format als benutzerdefinierte Parameter für Lambda zur Verfügung stellen.

So erstellen Sie einen S3 Object Lambda Access Point

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Object Lambda access points (Objekt-Lambda-Zugriffspunkte) die Option Create Object Lambda access point (Objekt-Lambda-Zugriffspunkt erstellen).
4. Geben Sie unter Name des Objekt-Lambda-Access Point den Namen ein, den Sie für den Object Lambda Access Point verwenden möchten (z. B. **tutorial-object-lambda-accesspoint**).
5. Geben Sie für Unterstützenden Zugangspunkt den Standardzugangspunkt ein, den Sie in [Schritt 3](#) erstellt haben (z. B. **tutorial-access-point**), oder navigieren Sie zu diesem Zugangspunkt und wählen dann Unterstützenden Zugangspunkt auswählen aus.
6. Um für S3 Objekte aus dem S3-Bucket abzurufen APIs, damit die Lambda-Funktion sie verarbeiten kann, wählen Sie GetObject.
7. Für die Funktion Lambda aufrufen können Sie eine der beiden folgenden Optionen für dieses Tutorial auswählen.
 - Wählen Sie Wählen Sie aus Funktionen in Ihrem Kontound wählen Sie die Lambda-Funktion aus, die Sie in [Schritt 4](#) (z. B. **tutorial-object-lambda-function**) aus der Lambda-Funktion-Dropdown-Liste erstellt haben.
 - Wählen Sie Enter ARN und geben Sie dann den Amazon-Ressourcennamen (ARN) der Lambda-Funktion ein, die Sie in [Schritt 4](#) erstellt haben.
8. Wählen Sie für Lambda-Funktionsversion \$ LATEST (die neueste Version der Lambda-Funktion, die Sie in [Schritt 4](#) erstellt haben).
9. (Optional) Wenn Sie Ihre Lambda-Funktion benötigen, um GET Anfragen mit Bereichs- und Artikelnummernüberschriften zu erkennen und zu verarbeiten, wählen Sie Lambda-Funktion

unterstützt Anfragen mit Bereich und Lambda-Funktion unterstützt Anfragen mit Artikelnummern. Deaktivieren Sie andernfalls diese beiden Kontrollkästchen.

Weitere Informationen zur Verwendung von Bereichs- oder Teilenummern mit S3 Object Lambda finden Sie unter [Arbeitet mit Range and partNumber Header](#).

10. (Optional) Fügen Sie unter Payload — optional JSON Text hinzu, um Ihre Lambda-Funktion mit zusätzlichen Informationen zu versorgen.

Eine Payload ist optionaler JSON Text, den Sie Ihrer Lambda-Funktion als Eingabe für alle Aufrufe bereitstellen können, die von einem bestimmten S3 Object Lambda Access Point kommen. Um das Verhalten für mehrere Object Lambda Access Points anzupassen, die dieselbe Lambda-Funktion aufrufen, können Sie Nutzlasten mit unterschiedlichen Parametern konfigurieren und so die Flexibilität Ihrer Lambda-Funktion erweitern.

Weitere Informationen zur Nutzlast finden Sie unter [Format und Verwendung des Ereigniskontexts](#).

11. (Optional) Wählen Sie für Anforderungsmetriken – optional die Option Deaktivieren oder Aktivieren aus, um Ihrem Object Lambda Access Point Amazon-S3-Überwachung hinzuzufügen. Anforderungsmetriken werden zum CloudWatch Standardtarif von Amazon abgerechnet. Weitere Informationen finden Sie unter [CloudWatchPreise](#).
12. Behalten Sie unter Objekt-Lambda-Zugriffspunkt-Richtlinie - optional die Standardeinstellung bei.

(Optional) Sie können eine Ressourcenrichtlinie festlegen. Diese Ressourcenrichtlinie gewährt die GetObject API Erlaubnis, den angegebenen Object Lambda Access Point zu verwenden.
13. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt und klicken Sie auf Erstellen eines Objekt-Lambda-Zugriffspunkts.

Schritt 7: Anzeigen der transformierten Daten

Jetzt ist S3 Object Lambda bereit, Ihre Daten für Ihren Anwendungsfall zu transformieren. In diesem Tutorial wandelt S3 Object Lambda den gesamten Text in Ihrem Objekt in Großbuchstaben um.

Teilschritte

- [Zeigen Sie die transformierten Daten in Ihrem S3 Object Lambda Access Point an](#)
- [Ausführen eines Python-Skripts zum Drucken der ursprünglichen und transformierten Daten](#)

Zeigen Sie die transformierten Daten in Ihrem S3 Object Lambda Access Point an

Wenn Sie den Abruf einer Datei über Ihren S3 Object Lambda Access Point anfordern, GetObject API rufen Sie S3 Object Lambda auf. S3 Object Lambda ruft die Lambda-Funktion auf, um Ihre Daten zu transformieren, und gibt dann die transformierten Daten als Antwort auf den Standard-S3-Aufruf zurück. GetObject API

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte den S3 Object Lambda Access Point aus, den Sie in [Schritt 6](#) erstellt haben (z. B. **tutorial-object-lambda-accesspoint**).
4. Wählen Sie auf der Registerkarte Objekte Ihres S3 Object Lambda Access Point die Datei mit dem gleichen Namen (z. B. `tutorial.txt`) wie diejenige aus, die Sie in [Schritt 2](#) in den S3-Bucket hochgeladen haben.

Diese Datei sollte alle transformierten Daten enthalten.

5. Um die transformierten Daten anzuzeigen, wählen Sie Öffnen oder Herunterladen aus.

Ausführen eines Python-Skripts zum Drucken der ursprünglichen und transformierten Daten

Sie können S3 Object Lambda mit Ihren vorhandenen Anwendungen verwenden. Aktualisieren Sie dazu Ihre Anwendungskonfiguration, sodass der neue S3 Object Lambda Access PointARN, den Sie in [Schritt 6](#) erstellt haben, zum Abrufen von Daten aus S3 verwendet wird.

Das folgende Python-Beispielskript druckt sowohl die ursprünglichen Daten aus dem S3-Bucket als auch die transformierten Daten aus dem S3 Object Lambda Access Point.

1. Loggen Sie sich ein in AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte die Optionsschaltfläche links neben dem S3 Object Lambda Access Point aus, den Sie in [Schritt 6](#) erstellt haben (z. B. **tutorial-object-lambda-accesspoint**).
4. Wählen Sie die Option CopyARN Kopieren aus.

5. Speichern Sie das ARN für eine spätere Verwendung.
6. Schreiben Sie ein Python-Skript auf Ihrem lokalen Computer, um sowohl die ursprünglichen Daten aus Ihrem S3-Bucket (z. B. `tutorial.txt`) als auch die transformierten Daten aus Ihrem S3 Object Lambda Access Point (z. B. `tutorial.txt`) zu drucken. Sie können folgendes Beispiel-Skript nutzen.

```
import boto3
from botocore.config import Config

s3 = boto3.client('s3', config=Config(signature_version='s3v4'))

def getObject(bucket, key):
    objectBody = s3.get_object(Bucket = bucket, Key = key)
    print(objectBody["Body"].read().decode("utf-8"))
    print("\n")

print('Original object from the S3 bucket:')
# Replace the two input parameters of getObject() below with
# the S3 bucket name that you created in Step 1 and
# the name of the file that you uploaded to the S3 bucket in Step 2
getObject("tutorial-bucket",
         "tutorial.txt")

print('Object transformed by S3 Object Lambda:')
# Replace the two input parameters of getObject() below with
# the ARN of your S3 Object Lambda Access Point that you saved earlier and
# the name of the file with the transformed data (which in this case is
# the same as the name of the file that you uploaded to the S3 bucket
# in Step 2)
getObject("arn:aws:s3-object-lambda:us-west-2:111122223333:accesspoint/tutorial-
object-lambda-accesspoint",
         "tutorial.txt")
```

7. Speichern Sie Ihr Python-Skript unter einem benutzerdefinierten Namen (z. B. `tutorial_print.py`) im Ordner (z. B. `object-lambda`), den Sie in [Schritt 4](#) auf Ihrem lokalen Computer erstellt haben.
8. Führen Sie in Ihrem lokalen Terminal den folgenden Befehl vom Stamm des Verzeichnisses aus (z. B. `object-lambda`), das Sie in [Schritt 4](#) erstellt haben.

```
python3 tutorial_print.py
```

Sie sollten sowohl die ursprünglichen Daten als auch die transformierten Daten (alle Texte in Großbuchstaben) über das Terminal sehen. z. B. sollten Sie etwas wie den folgenden Text sehen.

```
Original object from the S3 bucket:
```

```
Amazon S3 Object Lambda Tutorial:
```

```
You can add your own code to process data retrieved from S3 before  
returning it to an application.
```

```
Object transformed by S3 Object Lambda:
```

```
AMAZON S3 OBJECT LAMBDA TUTORIAL:
```

```
YOU CAN ADD YOUR OWN CODE TO PROCESS DATA RETRIEVED FROM S3 BEFORE  
RETURNING IT TO AN APPLICATION.
```

Schritt 8: Bereinigen

Wenn Sie Ihre Daten nur zu Lernzwecken mit S3 Object Lambda transformiert haben, löschen Sie die AWS Ressourcen, die Sie so zugewiesen haben, dass Ihnen keine Gebühren mehr anfallen.

Teilschritte

- [Löschen des Object Lambda Access Point](#)
- [Löschen des S3-Zugriffspunkts](#)
- [Löschen der Ausführungsrolle für Ihre Lambda-Funktion](#)
- [Löschen Sie die Lambda-Funktion](#)
- [Löschen Sie die CloudWatch Protokollgruppe](#)
- [Löschen Sie die Originaldatei im S3-Quell-Bucket](#)
- [Löschen des S3-Quell-Bucket](#)
- [Löschen Sie den IAM Benutzer](#)

Löschen des Object Lambda Access Point

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.

3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte die Optionsschaltfläche links neben dem S3 Object Lambda Access Point aus, den Sie in [Schritt 6](#) erstellt haben (z. B. **tutorial-object-lambda-accesspoint**).
4. Wählen Sie Delete (Löschen).
5. Bestätigen Sie, dass Sie Ihren Object Lambda Access Point löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Löschen wählen.

Löschen des S3-Zugriffspunkts

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Navigieren Sie zu dem Zugriffspunkt, den Sie in [Schritt 3](#) (z. B. **tutorial-access-point**) erstellt haben, und wählen Sie die Schaltfläche neben dem Namen des Zugriffspunkts aus.
4. Wählen Sie Löschen aus.
5. Bestätigen Sie, dass Sie Ihren Zugriffspunkt löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Löschen der Ausführungsrolle für Ihre Lambda-Funktion

1. Melden Sie sich bei AWS Management Console und öffne das AWS Lambda Konsole bei <https://console.aws.amazon.com/lambda/>.
2. Wählen Sie im linken Navigationsbereich die Option Funktionen aus.
3. Wählen Sie die Funktion aus, die Sie in [Schritt 4](#) erstellt haben (z. B. **tutorial-object-lambda-function**).
4. Wählen Sie auf der Detailseite Ihrer Lambda-Funktion den Tag Konfiguration und danach Berechtigungen im linken Navigationsbereich.
5. Wählen Sie unter Ausführungsrolle im Bereich den Link des Rollennamens aus. Die IAM Konsole wird geöffnet.
6. Wählen Sie auf der Übersichtsseite der IAM Konsole mit der Ausführungsrolle Ihrer Lambda-Funktion die Option Rolle löschen aus.
7. Klicken Sie im Dialogfeld auf Rolle löschen und anschließend auf Ja, löschen.

Löschen Sie die Lambda-Funktion

1. Im AWS Lambda Wählen Sie in der Konsole unter <https://console.aws.amazon.com/lambda/> Funktionen im linken Navigationsbereich aus.
2. Aktivieren Sie das Kontrollkästchen links neben dem Namen der Funktion, die Sie im Abschnitt [Schritt 4](#) (z. B. **tutorial-object-lambda-function**) erstellt haben.
3. Wählen Sie Actions (Aktionen) und anschließend Delete (Löschen).
4. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) die Option Delete (Löschen) aus.

Löschen Sie die CloudWatch Protokollgruppe

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im linken Navigationsbereich Log groups (Protokollgruppen) aus.
3. Suchen Sie die Protokollgruppe, deren Name mit der Lambda-Funktion endet, die Sie in [Schritt 4](#) (z. B. **tutorial-object-lambda-function**) erstellt haben.
4. Aktivieren Sie das Kontrollkästchen links neben dem Namen der Protokollgruppe.
5. Wählen Sie Actions (Aktionen) und dann Delete log group(s) (Protokollgruppe(n) löschen) aus.
6. Wählen Sie im Dialogfeld Delete log group(s) (Protokollgruppe(n) löschen) die Option Delete (Löschen) aus.

Löschen Sie die Originaldatei im S3-Quell-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Bucket-Name den Namen des Buckets aus, in den Sie die Originaldatei in [Schritt 2](#) (z. B. **tutorial-bucket**) hochgeladen haben.
4. Markieren Sie das Kontrollkästchen links neben dem Namen des Objekts, das Sie löschen möchten (z. B. `tutorial.txt`).
5. Wählen Sie Löschen.
6. Bestätigen Sie auf der Seite Löschen von Objekten im Abschnitt Objekte endgültig löschen?, dass Sie dieses Objekt löschen möchten, indem Sie **permanently delete** im Textfeld eingeben.

7. Wählen Sie Delete objects (Objekte löschen).

Löschen des S3-Quell-Bucket

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets die Optionsschaltfläche neben dem Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben.
4. Wählen Sie Löschen.
5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Löschen Sie den IAM Benutzer

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Benutzer, und aktivieren Sie dann das Kontrollkästchen neben dem Benutzer, den Sie löschen möchten.
3. Wählen Sie oben auf der Seite Delete (Löschen) aus.
4. Im Bereich Löschen **user name**? Geben Sie im Dialogfeld den Benutzernamen in das Texteingabefeld ein, um das Löschen des Benutzers zu bestätigen. Wählen Sie Löschen.

Nächste Schritte

Nach Abschluss dieses Tutorials können Sie die Lambda-Funktion an Ihren Anwendungsfall anpassen, um die von GET Standard-S3-Anfragen zurückgegebenen Daten zu ändern.

Im Folgenden finden Sie eine Liste der häufigsten Anwendungsfälle für S3 Object Lambda:

- Maskierung sensibler Daten für Sicherheit und Compliance.

Weitere Informationen finden Sie unter [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#).

- Filtern bestimmter Datenzeilen, um bestimmte Informationen bereitzustellen.
- Erweitern von Daten mit Informationen aus anderen Diensten oder Datenbanken.
- Konvertierung zwischen Datenformaten, z. B. Konvertierung in XML aus JSON Gründen der Anwendungskompatibilität.
- Komprimieren oder Dekomprimieren von Dateien, während sie heruntergeladen werden.
- Größenänderung und Wasserzeichen für Bilder.

Weitere Informationen finden Sie unter [Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen](#).

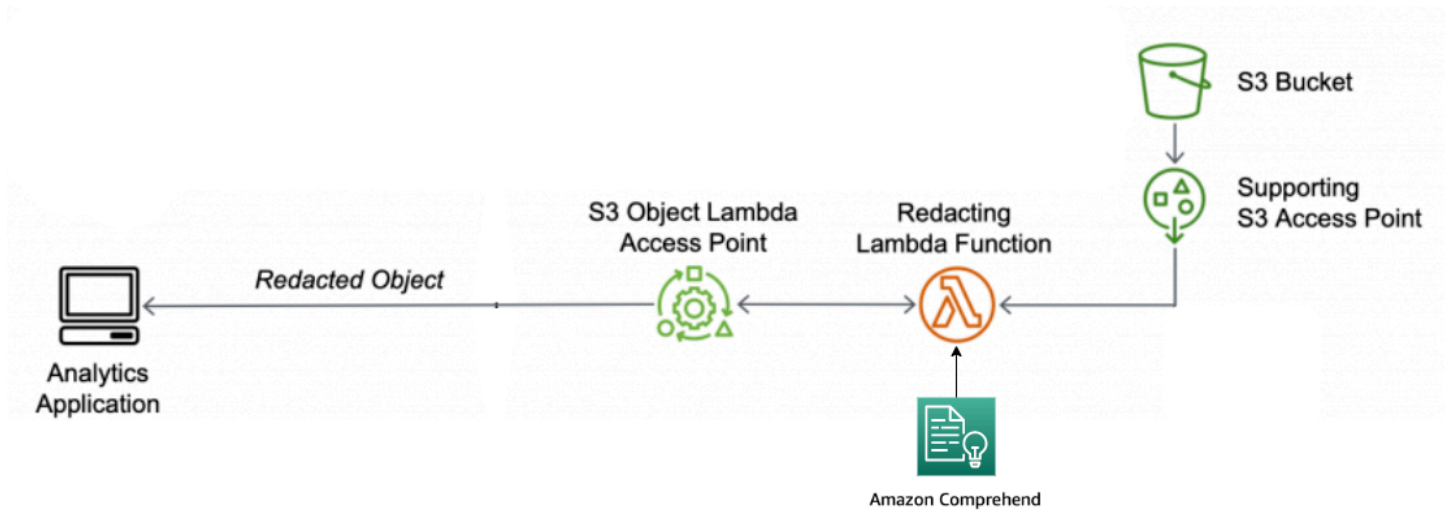
- Implementieren benutzerdefinierter Autorisierungsregeln für den Zugriff auf Daten.

Weitere Informationen zu S3 Object Lambda finden Sie unter [Transformieren von Objekten mit S3 Object Lambda](#).

Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend

Wenn Sie Amazon S3 für gemeinsam genutzte Datensätze verwenden, auf die mehrere Anwendungen und Benutzer zugreifen können, ist es wichtig, dass vertrauliche Informationen, wie z. B. personenbezogene Daten (PII), nur autorisierten Personen zur Verfügung stehen. Wenn eine Marketinganwendung beispielsweise Daten verwendet, die Daten enthalten PII, muss sie möglicherweise zuerst PII Daten maskieren, um die Datenschutzanforderungen zu erfüllen. Wenn eine Analytikanwendung ein Bestandsdatensatz für Produktionsaufträge verwendet, muss sie möglicherweise zuerst die Kreditkarteninformationen des Kunden überarbeiten, um unbeabsichtigte Datenverluste zu verhindern.

Mit [S3 Object Lambda](#) und einem vorgefertigten AWS Lambda Mit der von Amazon Comprehend unterstützten Funktion können Sie aus S3 abgerufene PII Daten schützen, bevor Sie sie an eine Anwendung zurückgeben. Insbesondere können Sie die vorgefertigte [Lambda-Funktion](#) als Redigierungsfunktion nutzen und sie an einen S3 Object Lambda Access Point anfügen. Wenn eine Anwendung (z. B. eine Analyseanwendung) [standardmäßige GET S3-Anfragen sendet, rufen diese über den S3](#) Object Lambda Access Point gestellten Anfragen die vordefinierte Lambda-Funktion zum Schwärzen auf, um PII Daten zu erkennen und zu redigieren, die aus einem S3-Bucket über einen unterstützenden S3-Zugriffspunkt abgerufen wurden. Dann gibt der S3 Object Lambda Access Point das redigierte Ergebnis zurück an die Anwendung aus.



Dabei verwendet die vordefinierte Lambda-Funktion [Amazon Comprehend](#), einen Service zur Verarbeitung natürlicher Sprache (NLP), um Variationen in der Darstellung zu erfassen, unabhängig davon, wie sie im Text PII vorhanden ist (z. B. numerisch oder als Kombination aus Wörtern und Zahlen). Amazon Comprehend kann sogar den Kontext im Text verwenden, um zu verstehen, ob es sich bei einer 4-stelligen Zahl um eine PIN, um die letzten vier Zahlen einer Sozialversicherungsnummer (SSN) oder um ein Jahr handelt. Amazon Comprehend verarbeitet jede Textdatei im UTF-8-Format und kann in großem Umfang schützen, ohne die PII Genauigkeit zu beeinträchtigen. Weitere Informationen finden Sie unter [Was ist Amazon Comprehend?](#) im Entwicklerhandbuch für Amazon Comprehend.

Ziel

In diesem Tutorial erfahren Sie, wie Sie S3 Object Lambda mit der vordefinierten Lambda-Funktion `ComprehendPiiRedactionS3ObjectLambda` verwenden. Diese Funktion verwendet Amazon Comprehend, um Entitäten zu erkennen. Diese Entitäten werden dann durch Sternchen ersetzt. Durch das PII Redigieren verbergen Sie sensible Daten, was zu Sicherheit und Compliance beitragen kann.

Sie lernen auch, wie Sie ein vorgefertigtes Gerät verwenden und konfigurieren AWS Lambda Funktion in der [AWS Serverless Application Repository](#) um mit S3 Object Lambda für eine einfache Bereitstellung zusammenzuarbeiten.

Themen

- [Voraussetzungen: Erstellen Sie einen IAM Benutzer mit Berechtigungen](#)
- [Schritt 1: Einen S3-Bucket erstellen](#)
- [Schritt 2: Hochladen einer Datei zu einem S3-Bucket](#)

- [Schritt 3: Erstellen eines S3-Zugriffspunkts](#)
- [Schritt 4: Konfigurieren und Bereitstellen einer vordefinierten Lambda-Funktion](#)
- [Schritt 5: Erstellen eines S3 Object Lambda Access Point](#)
- [Schritt 6: Verwenden des S3 Object Lambda Access Point zum Abrufen der redigierten Daten](#)
- [Schritt 7: Bereinigen](#)
- [Nächste Schritte](#)

Voraussetzungen: Erstellen Sie einen IAM Benutzer mit Berechtigungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie ein AWS Konto, mit dem Sie sich anmelden können AWS Identity and Access Management Benutzer (IAMBenutzer) mit den richtigen Berechtigungen.

Sie können einen IAM Benutzer für das Tutorial erstellen. Um dieses Tutorial abzuschließen, muss Ihr IAM Benutzer die folgenden IAM Richtlinien für den Zugriff auf relevante Dokumente beifügen AWS Ressourcen und bestimmte Aktionen ausführen.

Note

Der Einfachheit halber erstellt und verwendet dieses Tutorial einen IAM Benutzer. Denken Sie nach Abschluss dieses Tutorials an [Lösche den IAM Benutzer](#). Für den produktiven Einsatz empfehlen wir, dass Sie die [bewährten Sicherheitsmethoden IAM im IAM Benutzerhandbuch](#) befolgen. Eine bewährte Methode sieht vor, dass menschliche Benutzer für den Zugriff einen Verbund mit einem Identitätsanbieter verwenden AWS mit temporären Anmeldeinformationen. Eine weitere bewährte Methode besteht darin, zu verlangen, dass Workloads temporäre Anmeldeinformationen mit IAM Rollen für den Zugriff verwenden AWS. Um mehr über die Verwendung zu erfahren AWS IAM Identity Center Informationen zum Erstellen von Benutzern mit temporären Anmeldeinformationen finden Sie unter [Erste Schritte](#) im AWS IAM Identity Center Benutzerleitfaden.

In diesem Tutorial werden auch Richtlinien mit vollem Zugriff verwendet. Für die Verwendung in der Produktion empfehlen wir, dass Sie stattdessen nur die für Ihren Anwendungsfall erforderlichen Mindestberechtigungen gemäß [Bewährte Methoden in Bezug auf die Sicherheit](#) erteilen.

Ihr IAM Benutzer benötigt Folgendes AWS verwaltete Richtlinien:

- [AmazonS3 FullAccess — Erteilt](#) Berechtigungen für alle Amazon S3-Aktionen, einschließlich der Berechtigungen zum Erstellen und Verwenden eines Object Lambda Access Points.
- [AWSLambda_FullAccess](#) — Erteilt Berechtigungen für alle Lambda-Aktionen.
- [AWSCloudFormationFullAccess](#)— Erteilt allen Berechtigungen AWS CloudFormation Aktionen.
- [IAMFullAccess](#)— Erteilt Berechtigungen für alle IAM Aktionen.
- [IAMAccessAnalyzerReadOnlyAccess](#)— Erteilt Berechtigungen zum Lesen aller von Access Analyzer bereitgestellten IAM Zugriffsinformationen.

Sie können diese vorhandenen Richtlinien direkt anhängen, wenn Sie einen IAM Benutzer erstellen. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie im Benutzerhandbuch unter [IAMBenutzer erstellen \(Konsole\)](#). IAM

Darüber hinaus benötigt Ihr IAM Benutzer eine vom Kunden verwaltete Richtlinie. Um dem IAM Benutzer allen Benutzerberechtigungen zu gewähren AWS Serverless Application Repository Für Ressourcen und Aktionen müssen Sie eine IAM Richtlinie erstellen und die Richtlinie dem IAM Benutzer zuordnen.

Um eine IAM Richtlinie zu erstellen und an Ihren IAM Benutzer anzuhängen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich die Option Policies (Richtlinien) aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Tab Visual editor (Visueller Editor) für Service Choose a service (Wählen Sie einen Service) aus. Wählen Sie dann Serverless Application Repository.
5. Wählen Sie für Aktionen unter Manuelle Aktionen Alle Aktionen des Serverless Application Repository (serverlessrepo: *) für dieses Tutorial.

Als bewährte Methode für die Sicherheit sollten Sie basierend auf Ihrem Anwendungsfall nur für jene Aktionen und Ressourcen Berechtigungen zulassen, für die ein Benutzer Zugriff benötigt. Weitere Informationen finden Sie unter [Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch](#).

6. Wählen Sie für Ressourcen Alle Ressourcen für dieses Tutorial.

Als bewährte Methode sollten Sie Berechtigungen nur für bestimmte Ressourcen in bestimmten Konten definieren. Alternativ können Sie die geringste Berechtigung mit Bedingungsschlüsseln

erteilen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Geringste Zugriffsrechte gewähren](#).

7. Wählen Sie Next: Tags (Weiter: Tags) aus.
8. Klicken Sie auf Weiter: Prüfen.
9. Geben Sie auf der Seite Review policy (Richtlinie überprüfen) einen Namen (z. B. **tutorial-serverless-application-repository**) und eine Beschreibung (optional) für die Richtlinie ein, die Sie erstellen. Überprüfen Sie die Richtlinienzusammenfassung, um sicherzustellen, dass Sie die beabsichtigten Berechtigungen erteilt haben, und wählen Sie dann Richtlinie erstellen aus, um Ihre neue Richtlinie zu speichern.
10. Wählen Sie im linken Navigationsbereich Benutzer aus. Wählen Sie dann den IAM Benutzer für dieses Tutorial aus.
11. Wählen Sie auf der Seite Übersicht des ausgewählten Benutzers den Tab Berechtigungen und danach Hinzufügen von Berechtigungen.
12. Wählen Sie unter Berechtigungen gewähren die Option Vorhandene Richtlinien direkt anfügen aus.
13. Aktivieren Sie das Kontrollkästchen neben der gerade erstellten Richtlinie (z. B. **tutorial-serverless-application-repository**) und wählen Sie dann Nächstes: Überprüfung.
14. Prüfen Sie unter Richtlinienübersicht die Zusammenfassung, um sicherzustellen, dass Sie die beabsichtigte Richtlinie angefügt haben. Wählen Sie dann Add permissions (Berechtigungen hinzufügen) aus.

Schritt 1: Einen S3-Bucket erstellen

Erstellen Sie einen Bucket zum Speichern der ursprünglichen Daten, die Sie transformieren möchten.

So erstellen Sie einen Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

Die Seite Create bucket (Bucket erstellen) wird geöffnet.

4. Geben Sie für Bucket-Name einen Namen für Ihren Bucket ein (z. B. **tutorial-bucket**).

Weitere Informationen zur Benennung von Buckets in Amazon S3 finden Sie unter [Regeln für die Benennung von Buckets](#).

5. Wählen Sie für Region die AWS-Region wo sich der Bucket befinden soll.

Weitere Informationen zur Bucket-Region finden Sie unter [Bucket-Übersicht](#).

6. Belassen Sie die Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren bei den Standardeinstellungen (Alle öffentlichen Zugriffe blockieren ist aktiviert).

Es wird empfohlen, alle Einstellungen für öffentlichen Zugriff blockieren aktiviert zu belassen, es sei denn, Sie müssen eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

(Optional) Informationen zum Konfigurieren zusätzlicher Bucket-Einstellungen für Ihren spezifischen Anwendungsfall finden Sie unter [Erstellen eines Buckets](#).

8. Wählen Sie Bucket erstellen aus.

Schritt 2: Hochladen einer Datei zu einem S3-Bucket

Laden Sie eine Textdatei mit bekannten PII Daten verschiedener Typen, wie Namen, Bankinformationen, Telefonnummern usw., als Originaldaten in den S3-Bucket hochSSNs, die Sie später in diesem Tutorial PII redigieren werden.

Sie können beispielsweise nach der `tutorial.txt`-Datei hochladen. Dies ist ein Beispiel für eine Eingabedatei von Amazon Comprehend.

```
Hello Zhang Wei, I am John. Your AnyCompany Financial Services,
LLC credit card account 1111-0000-1111-0008 has a minimum payment
of $24.53 that is due by July 31st. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account number XXXXXX1111 with the routing number XXXXX0000.
```

```
Your latest statement was mailed to 100 Main Street, Any City,
WA 98121.
```

```
After your payment is received, you will receive a confirmation
text message at 206-555-0100.
```

```
If you have questions about your bill, AnyCompany Customer Service
```

is available by phone at 206-555-0199 or email at support@anycompany.com.

Hochladen einer Datei in einen Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben, in den Sie die Datei hochladen möchten.
4. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
5. Wählen Sie auf der Seite Upload unter Dateien und Ordner die Option Dateien hinzufügen aus.
6. Wählen Sie eine hochzuladende Datei und dann Open (Öffnen) aus. Sie können z. B. das oben erwähnte tutorial.txt-Dateibeispiel hochladen.
7. Klicken Sie auf Hochladen.

Schritt 3: Erstellen eines S3-Zugriffspunkts

Um einen S3 Object Lambda Access Point für den Zugriff und die Umwandlung der ursprünglichen Daten zu verwenden, müssen Sie einen S3-Zugriffspunkt erstellen und dem S3-Bucket zuordnen, den Sie in [Schritt 1](#) erstellt haben. Der Access Point muss sich im selben befinden AWS-Region wie die Objekte, die Sie transformieren möchten.

Später in diesem Tutorial verwenden Sie diesen Zugriffspunkt als unterstützenden Zugriffspunkt für Ihren Object Lambda Access Point.

So erstellen Sie einen Zugangspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Wählen Sie auf der Seite Zugriffspunkte die Option Zugriffspunkt erstellen aus.
4. Geben Sie im Feld Name des Zugriffspunkts den gewünschten Namen für den Zugriffspunkt ein (z. B. **tutorial-pii-access-point**).

Weitere Hinweise zur Benennung von Zugangspunkten finden Sie unter [Regeln zur Benennung von Amazon S3-Zugriffspunkten](#).

5. Wählen Sie im Feld Bucket-Name den Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben. S3 hängt den Zugriffspunkt an diesen Bucket an.

(Optional) Sie können S3 durchsuchen auswählen, um die Buckets in Ihrem Konto zu browsen und zu durchsuchen. Wenn Sie Browse S3 (S3 durchsuchen) wählen, wählen Sie den gewünschten Bucket aus und dann Choose path (Pfad wählen), um das Feld Bucket name (Bucket-Name) mit dem Namen dieses Buckets zu füllen.

6. Wählen Sie für Netzwerkursprung Internetaus.

Weitere Informationen zu Netzwerkursprüngen für Zugangspunkte finden Sie unter [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#).

7. Alle Block-Public-Access-Einstellungen sind standardmäßig für Zugriffspunkte aktiviert. Sie sollten alle Einstellungen für das Blockieren aller öffentlichen Zugriffe aktiviert lassen. Weitere Informationen finden Sie unter [Verwalten des öffentlichen Zugriffs auf Zugriffspunkte](#).
8. Behalten Sie für alle anderen Zugriffspunkteinstellungen die Standardeinstellungen bei.

(Optional) Sie können die Zugriffspunkteinstellungen ändern, um Ihren Anwendungsfall zu unterstützen. Für dieses Tutorial empfehlen wir, dass die Standardeinstellungen beibehalten werden.

(Optional) Wenn Sie den Zugriff auf Ihren Zugriffspunkt verwalten müssen, können Sie eine Zugriffspunkt-Richtlinie angeben. Weitere Informationen finden Sie unter [Beispiele von - Zugriffspunktrichtlinien](#).

9. Wählen Sie Zugangspunkt erstellen aus.

Schritt 4: Konfigurieren und Bereitstellen einer vordefinierten Lambda-Funktion

Um PII Daten zu redigieren, konfigurieren und implementieren Sie das vorgefertigte AWS Lambda Funktion ComprehendPiiRedactionS3ObjectLambda zur Verwendung mit Ihrem S3 Object Lambda Access Point.

Konfigurieren und Bereitstellen der Lambda-Funktion

1. Loggen Sie sich ein bei AWS Management Console und sehen Sie sich die [ComprehendPiiRedactionS3ObjectLambda](#) Funktion im AWS Serverless Application Repository.
2. Behalten Sie für Anwendungseinstellungen unter Anwendungsname den Standardwert (ComprehendPiiRedactionS3ObjectLambda) für dieses Tutorial bei.

(Optional) Sie können den Namen eingeben, den Sie dieser Anwendung geben möchten. Sie können dies tun, wenn Sie planen, mehrere Lambda-Funktionen für unterschiedliche Zugriffsanforderungen für denselben freigegebenen Datensatz zu konfigurieren.
3. Behalten Sie für MaskCharacter den Standardwert (*) bei. Das Maskenzeichen ersetzt jedes Zeichen in der geschwärzten PII Entität.
4. Behalten Sie für MaskModed den Standardwert (MASK) bei. Der MaskModeWert gibt an, ob die PII Entität mit dem MASK Zeichen oder dem PII_ENTITY_TYPE Wert geschwärzt wird.
5. Um die angegebenen Datentypen zu schwärzen PiiEntityTypes, behalten Sie für den Standardwert bei. ALL Der PiiEntityTypesWert gibt die PII Entitätstypen an, die bei der Schwärzung berücksichtigt werden sollen.

Weitere Informationen zur Liste der unterstützten PII Entitätstypen finden Sie unter [Detect Personally Identifiable Information \(PII\)](#) im Amazon Comprehend Developer Guide.

6. Lassen Sie die restlichen Einstellungen bei den Standardwerten.

(Optional) Informationen zum Konfigurieren von zusätzlichen Einstellungen für Ihren speziellen Anwendungsfall finden Sie im Abschnitt Readme-Datei auf der linken Seite der Seite.
7. Aktivieren Sie das Kontrollkästchen neben Ich bestätige, dass diese App benutzerdefinierte IAM Rollen erstellt.
8. Wählen Sie Bereitstellen.
9. Wählen Sie auf der Seite der neuen Anwendung unter Ressourcen die Logische ID der Lambda-Funktion, die Sie bereitgestellt haben, um die Funktion auf der Lambda-Funktionsseite zu überprüfen.

Schritt 5: Erstellen eines S3 Object Lambda Access Point

Ein S3 Object Lambda Access Point bietet die Flexibilität, eine Lambda-Funktion direkt von einer GET S3-Anfrage aus aufzurufen, sodass die Funktion PII Daten, die von einem S3-Zugangspunkt

abgerufen wurden, unkenntlich machen kann. Beim Erstellen und Konfigurieren eines S3 Object Lambda Access Points müssen Sie die schwärzende Lambda-Funktion angeben, um den Ereigniskontext aufzurufen und im JSON Format als benutzerdefinierte Parameter bereitzustellen, die Lambda verwenden kann.

Der Ereigniskontext stellt Informationen über die Anforderung bereit, die in dem von S3 Object Lambda an Lambda übergebenen Ereignis gestellt wird. Weitere Informationen zu allen Feldern im Ereigniskontext finden Sie unter [Format und Verwendung des Ereigniskontexts](#).

So erstellen Sie einen S3 Object Lambda Access Point

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Object Lambda access points (Objekt-Lambda-Zugriffspunkte) die Option Create Object Lambda access point (Objekt-Lambda-Zugriffspunkt erstellen).
4. Geben Sie unter Name des Objekt-Lambda-Access Point den Namen ein, den Sie für den Object Lambda Access Point verwenden möchten (z. B. **tutorial-pii-object-lambda-accesspoint**).
5. Geben Sie für Unterstützenden Zugangspunkt den Standardzugangspunkt ein, den Sie in [Schritt 3](#) erstellt haben (z. B. **tutorial-pii-access-point**), oder navigieren Sie zu diesem Zugangspunkt und wählen dann Unterstützenden Zugangspunkt auswählen aus.
6. Um für S3 Objekte aus dem S3-Bucket abzurufen APIs, damit die Lambda-Funktion sie verarbeiten kann, wählen Sie GetObject.
7. Für die Funktion Lambda aufrufen können Sie eine der beiden folgenden Optionen für dieses Tutorial auswählen.
 - Wählen Sie Wählen Sie aus Funktionen in Ihrem Kontound wählen Sie die Lambda-Funktion aus, die Sie in [Schritt 4](#) (z. B. **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**) aus der Lambda-Funktion-Dropdown-Liste bereitgestellt haben.
 - Wählen Sie Enter ARN und geben Sie dann den Amazon-Ressourcennamen (ARN) der Lambda-Funktion ein, die Sie in [Schritt 4](#) erstellt haben.
8. Wählen Sie für Lambda-Funktionsversion \$ aus LATEST (die neueste Version der Lambda-Funktion, die Sie in [Schritt 4](#) bereitgestellt haben).
9. (Optional) Wenn Sie Ihre Lambda-Funktion benötigen, um GET Anfragen mit Bereichs- und Artikelnummernüberschriften zu erkennen und zu verarbeiten, wählen Sie Lambda-Funktion

unterstützt Anfragen mit Bereich und Lambda-Funktion unterstützt Anfragen mit Artikelnummern. Deaktivieren Sie andernfalls diese beiden Kontrollkästchen.

Weitere Informationen zur Verwendung von Bereichs- oder Teilenummern mit S3 Object Lambda finden Sie unter [Arbeitet mit Range and partNumber Header](#).

10. (Optional) Fügen Sie unter Payload — optional JSON Text hinzu, um Ihre Lambda-Funktion mit zusätzlichen Informationen zu versorgen.

Eine Payload ist optionaler JSON Text, den Sie Ihrer Lambda-Funktion als Eingabe für alle Aufrufe bereitstellen können, die von einem bestimmten S3 Object Lambda Access Point kommen. Um das Verhalten für mehrere Object Lambda Access Points anzupassen, die dieselbe Lambda-Funktion aufrufen, können Sie Nutzlasten mit unterschiedlichen Parametern konfigurieren und so die Flexibilität Ihrer Lambda-Funktion erweitern.

Weitere Informationen zur Nutzlast finden Sie unter [Format und Verwendung des Ereigniskontexts](#).

11. (Optional) Wählen Sie für Anforderungsmetriken – optional die Option Deaktivieren oder Aktivieren aus, um Ihrem Object Lambda Access Point Amazon-S3-Überwachung hinzuzufügen. Anforderungsmetriken werden zum CloudWatch Standardtarif von Amazon abgerechnet. Weitere Informationen finden Sie unter [CloudWatchPreise](#).
12. Behalten Sie unter Objekt-Lambda-Zugriffspunkt-Richtlinie - optional die Standardeinstellung bei.
(Optional) Sie können eine Ressourcenrichtlinie festlegen. Diese Ressourcenrichtlinie gewährt die GetObject API Erlaubnis, den angegebenen Object Lambda Access Point zu verwenden.
13. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt und klicken Sie auf Erstellen eines Objekt-Lambda-Zugriffspunkts.

Schritt 6: Verwenden des S3 Object Lambda Access Point zum Abrufen der redigierten Daten

Jetzt ist S3 Object Lambda bereit, PII Daten aus Ihrer Originaldatei zu redigieren.

So verwenden Sie den S3 Object Lambda Access Point zum Abrufen der redigierten Daten

Wenn Sie den Abruf einer Datei über Ihren S3 Object Lambda Access Point anfordern, GetObject API rufen Sie S3 Object Lambda auf. S3 Object Lambda ruft die Lambda-Funktion auf, um Ihre PII

Daten zu redigieren, und gibt die transformierten Daten als Antwort auf den Standard-S3-Aufruf zurück. GetObject API

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte den S3 Object Lambda Access Point aus, den Sie in [Schritt 5](#) erstellt haben (z. B. **tutorial-pii-object-lambda-accesspoint**).
4. Wählen Sie auf der Registerkarte Objekte Ihres S3 Object Lambda Access Point die Datei mit dem gleichen Namen (z. B. `tutorial.txt`) wie diejenige aus, die Sie in [Schritt 2](#) in den S3-Bucket hochgeladen haben.

Diese Datei sollte alle transformierten Daten enthalten.

5. Um die transformierten Daten anzuzeigen, wählen Sie Öffnen oder Herunterladen aus.

Sie sollten in der Lage sein, die redigierte Datei zu sehen, wie im folgenden Beispiel gezeigt.

```
Hello *****. Your AnyCompany Financial Services,
LLC credit card account ***** has a minimum payment
of $24.53 that is due by *****. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account ***** with the routing number *****.

Your latest statement was mailed to *****.
After your payment is received, you will receive a confirmation
text message at *****.
If you have questions about your bill, AnyCompany Customer Service
is available by phone at ***** or
email at *****.
```

Schritt 7: Bereinigen

Wenn Sie Ihre Daten nur zu Lernzwecken mit S3 Object Lambda redigiert haben, löschen Sie die AWS Ressourcen, die Sie so zugewiesen haben, dass Ihnen keine Gebühren mehr anfallen.

Teilschritte

- [Löschen des Object Lambda Access Point](#)
- [Löschen des S3-Zugriffspunkts](#)
- [Löschen Sie die Lambda-Funktion](#)
- [Löschen Sie die CloudWatch Protokollgruppe](#)
- [Löschen Sie die Originaldatei im S3-Quell-Bucket](#)
- [Löschen des S3-Quell-Bucket](#)
- [Löschen Sie die IAM Rolle für Ihre Lambda-Funktion](#)
- [Löschen Sie die vom Kunden verwaltete Richtlinie für Ihren IAM Benutzer](#)
- [Lösche den IAM Benutzer](#)

Löschen des Object Lambda Access Point

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte die Optionsschaltfläche links neben dem S3 Object Lambda Access Point aus, den Sie in [Schritt 5](#) erstellt haben (z. B. **tutorial-pii-object-lambda-accesspoint**).
4. Wählen Sie Delete (Löschen).
5. Bestätigen Sie, dass Sie Ihren Object Lambda Access Point löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Löschen wählen.

Löschen des S3-Zugriffspunkts

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Navigieren Sie zu dem Zugriffspunkt, den Sie in [Schritt 3](#) (z. B. **tutorial-pii-accesspoint**) erstellt haben, und wählen Sie die Optionsschaltfläche neben dem Namen des Zugriffspunkts aus.
4. Wählen Sie Löschen.
5. Bestätigen Sie, dass Sie Ihren Zugriffspunkt löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Löschen Sie die Lambda-Funktion

1. Im AWS Lambda Konsole unter <https://console.aws.amazon.com/lambda/>, wählen Sie im linken Navigationsbereich Funktionen aus.
2. Wählen Sie die Funktion aus, die Sie in [Schritt 4](#) (z. B. **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**) erstellt haben.
3. Wählen Sie Aktionen und anschließend Löschen aus.
4. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) die Option Delete (Löschen) aus.

Löschen Sie die CloudWatch Protokollgruppe

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im linken Navigationsbereich Log groups (Protokollgruppen) aus.
3. Suchen Sie die Protokollgruppe, deren Name mit der Lambda-Funktion endet, die Sie in [Schritt 4](#) (z. B. **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**) erstellt haben.
4. Wählen Sie Actions (Aktionen) und dann Delete log group(s) (Protokollgruppe(n) löschen) aus.
5. Wählen Sie im Dialogfeld Delete log group(s) (Protokollgruppe(n) löschen) die Option Delete (Löschen) aus.

Löschen Sie die Originaldatei im S3-Quell-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Bucket-Name den Namen des Buckets aus, in den Sie die Originaldatei in [Schritt 2](#) (z. B. **tutorial-bucket**) hochgeladen haben.
4. Markieren Sie das Kontrollkästchen links neben dem Namen des Objekts, das Sie löschen möchten (z. B. `tutorial.txt`).
5. Wählen Sie Löschen.
6. Bestätigen Sie auf der Seite Löschen von Objekten im Abschnitt Objekte endgültig löschen?, dass Sie dieses Objekt löschen möchten, indem Sie **permanently delete** im Textfeld eingeben.
7. Wählen Sie Delete objects (Objekte löschen).

Löschen des S3-Quell-Bucket

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben.
4. Wählen Sie Löschen.
5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Löschen Sie die IAM Rolle für Ihre Lambda-Funktion

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen), und aktivieren Sie dann das Kontrollkästchen neben der Rolle, die Sie löschen möchten. Der Rollename beginnt mit dem Namen der Lambda-Funktion, die Sie in [Schritt 4](#) (z. B. **serverlessrepo-ComprehendPiiRedactionS3ObjectLambda**) erstellt haben.
3. Wählen Sie Löschen.
4. Geben Sie im Dialogfeld Löschen den Rollennamen in das Texteingabefeld ein, um das Löschen zu bestätigen. Wählen Sie dann Löschen aus.

Löschen Sie die vom Kunden verwaltete Richtlinie für Ihren IAM Benutzer

1. Melden Sie sich bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich die Option Richtlinien aus.
3. Geben Sie auf der Seite Richtlinien den Namen der vom Kunden verwalteten Richtlinie ein, die Sie in [Voraussetzungen](#) (z. B. **tutorial-serverless-application-repository**) erstellt haben, in das Suchfeld ein, um die Richtlinienliste zu filtern. Aktivieren Sie die Optionsschaltfläche neben dem Namen der Richtlinie, die Sie löschen möchten.
4. Wählen Sie Aktionen und anschließend Löschen aus.

5. Bestätigen Sie, dass Sie diese Richtlinie löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Lösche den IAM Benutzer

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Benutzer, und aktivieren Sie dann das Kontrollkästchen neben dem Benutzer, den Sie löschen möchten.
3. Wählen Sie oben auf der Seite Delete (Löschen) aus.
4. Im Bereich Löschen **user name**? Geben Sie im Dialogfeld den Benutzernamen in das Texteingabefeld ein, um das Löschen des Benutzers zu bestätigen. Wählen Sie Löschen.

Nächste Schritte

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie die folgenden verwandten Anwendungsfälle näher untersuchen:

- Sie können mehrere S3 Object Lambda Access Points erstellen und sie mit vorgefertigten Lambda-Funktionen aktivieren, die PII je nach den Geschäftsanforderungen der Datenzugriff unterschiedlich konfiguriert sind, um bestimmte Typen zu redigieren.

Jeder Benutzertyp nimmt eine IAM Rolle an und hat nur Zugriff auf einen S3 Object Lambda Access Point (verwaltet durch IAM Richtlinien). Dann fügen Sie jede `ComprehendPiiRedactionS3ObjectLambda`-Lambda-Funktion, die für einen anderen Redigierungsanwendungsfall konfiguriert ist, an einen anderen S3 Object Lambda Access Point an. Für jeden S3 Object Lambda Access Point können Sie über einen unterstützenden S3-Zugriffspunkt verfügen, um Daten aus einem S3-Bucket zu lesen, der den freigegebenen Datensatz speichert.

Weitere Informationen zum Erstellen einer S3-Bucket-Richtlinie, die es Benutzern ermöglicht, nur über S3-Zugriffspunkte aus dem Bucket zu lesen, finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#).

Weitere Informationen dazu, wie Sie einem Benutzer die Berechtigung für den Zugriff auf die Lambda-Funktion, den S3-Zugriffspunkt und den S3 Object Lambda Access Point erteilen, finden Sie unter [Konfiguration von IAM Richtlinien für Object Lambda Access Points](#).

- Sie können Ihre eigene Lambda-Funktion erstellen und S3 Object Lambda mit Ihrer angepassten Lambda-Funktion verwenden, um Ihre spezifischen Datenanforderungen zu erfüllen.

Um beispielsweise verschiedene Datenwerte zu untersuchen, können Sie S3 Object Lambda und Ihre eigene Lambda-Funktion verwenden, die zusätzliche [Amazon-Comprehend-Funktionen](#) verwenden, z. B. Entitätserkennung, Schlüsselphrasenerkennung, Stimmungsanalyse und Dokumentklassifizierung, um Daten zu verarbeiten. Sie können S3 Object Lambda auch zusammen mit [Amazon Comprehend Medical](#), einem HIPAA geeigneten NLP Service, verwenden, um Daten kontextsensitiv zu analysieren und zu extrahieren.

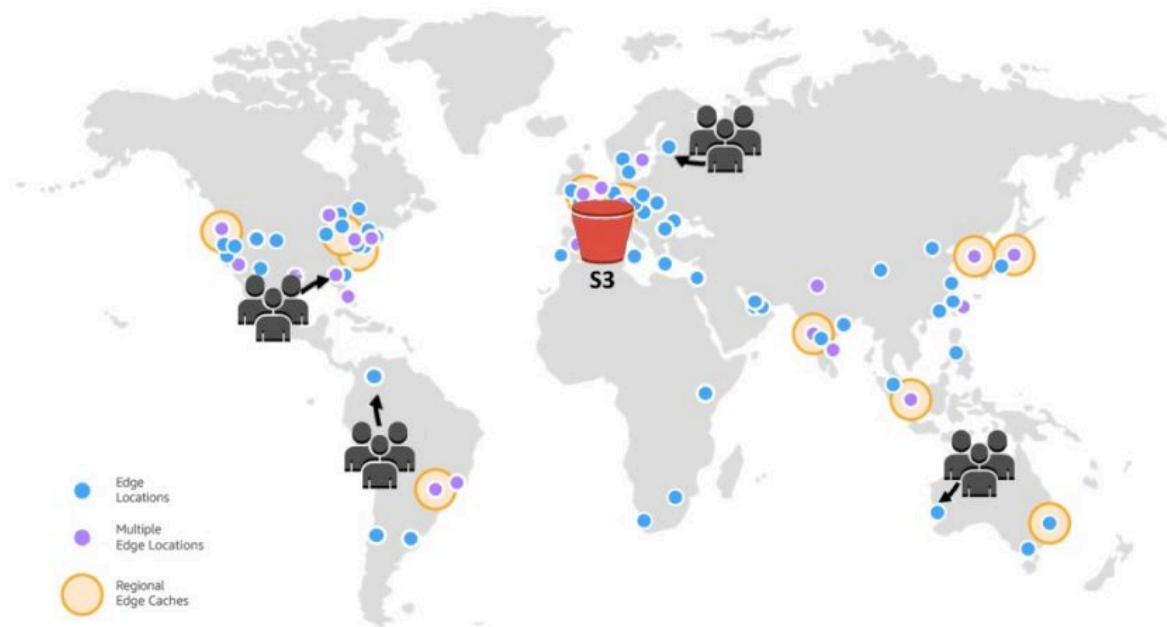
Weitere Informationen zum Transformieren von Daten mit S3 Object Lambda und Ihrer eigenen Lambda-Funktion finden Sie unter [Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#).

Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53

Sie können Amazon S3 mit Amazon verwenden CloudFront , um Videos für die On-Demand-Ansicht auf sichere und skalierbare Weise zu hosten. Video-on-Demand-Streaming (VOD) bedeutet, dass Ihre Videoinhalte auf einem Server gespeichert werden und Zuschauer sie jederzeit ansehen können.

CloudFront ist ein schneller, hochsicherer und programmierbarer Content Delivery Network (CDN) - Dienst. CloudFront kann Ihre Inhalte sicher HTTPS von allen CloudFront Edge-Standorten auf der ganzen Welt aus bereitstellen. Weitere Informationen zu CloudFront finden Sie unter [Was ist Amazon CloudFront?](#) im Amazon CloudFront Developer Guide.

CloudFront Caching reduziert die Anzahl der Anfragen, auf die Ihr Ursprungsserver direkt antworten muss. Wenn ein Zuschauer (Endbenutzer) ein Video anfordert, das Sie bereitstellen CloudFront, wird die Anfrage an einen Standort in der Nähe des Randes weitergeleitet, der sich näher am Standort des Betrachters befindet. CloudFront stellt das Video aus seinem Cache bereit und ruft es nur dann aus dem S3-Bucket ab, wenn es noch nicht zwischengespeichert ist. Diese Caching-Verwaltungsfunktion beschleunigt die Bereitstellung Ihres Videos für Viewer weltweit mit geringer Latenz, hohem Durchsatz und hohen Übertragungsgeschwindigkeiten. Weitere Informationen zur CloudFront Cachingverwaltung finden Sie unter [Optimieren von Caching und Verfügbarkeit](#) im Amazon CloudFront Developer Guide.



Ziel

In diesem Tutorial konfigurieren Sie einen S3-Bucket CloudFront für das Hosten von On-Demand-Videostreaming mit Amazon Route 53 für Domain Name System (DNS) und benutzerdefinierter Domainverwaltung.

Themen

- [Voraussetzungen: Registrieren und Konfigurieren einer benutzerdefinierten Domäne mit Route 53](#)
- [Schritt 1: Einen S3-Bucket erstellen](#)
- [Schritt 2: Hochladen eines Videos in den S3-Bucket](#)
- [Schritt 3: Erstellen Sie eine ursprüngliche CloudFront Zugriffsidentität](#)
- [Schritt 4: Erstellen Sie eine CloudFront Distribution](#)
- [Schritt 5: Greifen Sie über die CloudFront Distribution auf das Video zu](#)
- [Schritt 6: Konfigurieren Sie Ihre CloudFront Distribution so, dass sie Ihren benutzerdefinierten Domainnamen verwendet](#)
- [Schritt 7: Greifen Sie über die CloudFront Distribution mit dem benutzerdefinierten Domainnamen auf das S3-Video zu](#)

- [\(Optional\) Schritt 8: Zeigen Sie Daten zu Anfragen an, die bei Ihrer CloudFront Distribution eingegangen sind](#)
- [Schritt 9: Bereinigen](#)
- [Nächste Schritte](#)

Voraussetzungen: Registrieren und Konfigurieren einer benutzerdefinierten Domäne mit Route 53

Bevor Sie mit diesem Tutorial beginnen, müssen Sie eine benutzerdefinierte Domain (z. B. **example.com**) mit Route 53 registrieren und konfigurieren, damit Sie Ihre CloudFront Distribution so konfigurieren können, dass sie später einen benutzerdefinierten Domainnamen verwendet.

Ohne einen benutzerdefinierten Domainnamen ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront einen Server gehostetURL, der wie folgt aussieht:

```
https://CloudFront distribution domain name/Path to an S3 video
```

Beispiel, **https://d111111abcdef8.cloudfront.net/sample.mp4**.

Nachdem Sie Ihre CloudFront Distribution für die Verwendung eines mit Route 53 konfigurierten benutzerdefinierten Domainnamens konfiguriert haben, ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront eine URL gehostetURL, die etwa wie folgt aussieht:

```
https://CloudFront distribution alternate domain name/Path to an S3 video
```

Beispiel, **https://www.example.com/sample.mp4**. Ein benutzerdefinierter Domänenname ist für die Viewer einfacher und intuitiver zu verwenden.

Informationen zur Anmeldung einer benutzerdefinierten Domäne finden Sie unter [Registrieren einer neuen Domäne mit Route 53](#) im Amazon Route 53-Entwicklerhandbuch.

Wenn Sie einen Domänennamen bei Route 53 registrieren, erstellt Route 53 die gehostete Zone für Sie, die Sie später in diesem Tutorial verwenden werden. In dieser gehosteten Zone speichern Sie Informationen darüber, wie Sie den Traffic für Ihre Domain weiterleiten, z. B. an eine EC2 Amazon-Instance oder eine CloudFront Distribution.

Im Zusammenhang mit der Domainregistrierung, Ihrer Hosting-Zone und DNS Anfragen, die über Ihre Domain eingehen, fallen Gebühren an. Weitere Informationen dazu finden Sie unter [Amazon Route 53 – Preise](#).

Note

Wenn Sie eine Domäne registrieren, kostet diese sofort Geld. Dies kann nicht rückgängig gemacht werden. Sie können wählen, die Domäne nicht automatisch zu erneuern, aber Sie bezahlen im Voraus und besitzen sie für das Jahr. Weitere Informationen finden Sie unter [Registrieren einer neuen Domäne](#) im Amazon-Route-53-Entwicklerhandbuch.

Schritt 1: Einen S3-Bucket erstellen

Erstellen Sie einen Bucket, um das Originalvideo zu speichern, das Sie streamen möchten.

So erstellen Sie einen Bucket

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

Die Seite Bucket erstellen wird geöffnet.

4. Geben Sie im Feld Bucket Name einen Namen für Ihren Bucket ein (z. B. **tutorial-bucket**).

Weitere Informationen zur Benennung von Buckets in Amazon S3 finden Sie unter [Regeln für die Benennung von Buckets](#).

5. Wählen Sie für Region die AWS-Region wo sich der Bucket befinden soll.

Wenn möglich, sollten Sie die Region auswählen, die der Mehrheit Ihrer Viewer am nächsten ist. Weitere Informationen zur Bucket-Region finden Sie unter [Bucket-Übersicht](#).

6. Belassen Sie die Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren bei den Standardeinstellungen (Alle öffentlichen Zugriffe blockieren ist aktiviert).

Auch wenn „Allen öffentlichen Zugriff blockieren“ aktiviert ist, können Zuschauer weiterhin auf das hochgeladene Video zugreifen CloudFront. Diese Funktion ist ein großer Vorteil beim Hosten eines in S3 gespeicherten Videos. CloudFront

Es wird empfohlen, alle Einstellungen aktiviert zu belassen, es sei denn, Sie müssen eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

(Optional) Informationen zum Konfigurieren zusätzlicher Bucket-Einstellungen für Ihren spezifischen Anwendungsfall finden Sie unter [Erstellen eines Buckets](#).

8. Wählen Sie Bucket erstellen aus.

Schritt 2: Hochladen eines Videos in den S3-Bucket

Im folgenden Verfahren wird beschrieben, wie Sie eine Videodatei mithilfe der Konsole in einen S3-Bucket hochladen. Wenn Sie ein Video in S3 hochladen, können Sie auch [Amazon S3 Transfer Acceleration](#) nutzen, um schnelle und sichere Dateiübertragungen zu konfigurieren. Transfer Acceleration kann das Hochladen von Videos zu Ihrem S3-Bucket zur Übertragung größerer Videos über große Entfernungen beschleunigen. Weitere Informationen finden Sie unter [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#).

So laden Sie die Datei in einen Bucket hoch

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben, in den Sie die Datei hochladen möchten.
4. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
5. Wählen Sie auf der Seite Upload unter Dateien und Ordner die Option Dateien hinzufügen aus.
6. Wählen Sie eine hochzuladende Datei und dann Öffnen aus.

Sie können beispielsweise eine Videodatei mit dem Namen `sample.mp4` hochladen.

7. Klicken Sie auf Hochladen.

Schritt 3: Erstellen Sie eine ursprüngliche CloudFront Zugriffsidetität

Um den direkten Zugriff auf das Video von Ihrem S3-Bucket aus einzuschränken, erstellen Sie einen speziellen CloudFront Benutzer, eine sogenannte Origin-Zugriffsidetität (OAI). Sie werden das später in diesem Tutorial OAI mit Ihrer Distribution verknüpfen. Durch die Verwendung von stellen Sie sicher OAI, dass Zuschauer das Video nicht umgehen CloudFront und direkt aus dem S3-Bucket abrufen können. Nur CloudFront OAI sie können auf die Datei im S3-Bucket zugreifen. Weitere Informationen finden Sie unter [Beschränken des Zugriffs auf Amazon S3 S3-Inhalte mithilfe von OAI](#) im Amazon CloudFront Developer Guide.

Um eine zu erstellen CloudFront OAI

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Sicherheit Ursprungszugriff aus.
3. Wählen Sie unter der Registerkarte Identitäten Ursprungszugriffsidetität erstellen aus.
4. Geben Sie einen Namen ein (z. B. **S3-OAI**) als die neue Ursprungszugriffsidetität ein.
5. Wählen Sie Create (Erstellen) aus.

Schritt 4: Erstellen Sie eine CloudFront Distribution

Um das Video in Ihrem S3-Bucket bereitzustellen und zu verteilen, müssen Sie eine CloudFront Distribution erstellen. CloudFront

Teilschritte

- [Erstellen Sie eine CloudFront Distribution](#)
- [Überprüfen der Bucket-Richtlinie](#)

Erstellen Sie eine CloudFront Distribution

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Wählen Sie Verteilung erstellen aus.

4. Wählen Sie im Abschnitt Ursprung für Ursprungsdomäne den Domänennamen Ihres S3-Ursprungs aus, der mit dem Namen des S3-Buckets beginnt, den Sie unter [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben.
5. Wählen Sie für Ursprungszugriff die Option Legacy-Zugriffsidentitäten aus.
6. Wählen Sie unter Ursprungszugriffsidentität die Ursprungszugriffsidentität aus, die Sie in [Schritt 3](#) (z. B. **S3-OAI**) erstellt haben.
7. Wählen Sie unter Bucket-Richtlinie die Option Ja, Bucket-Richtlinie aktualisieren aus.
8. Wählen Sie im Abschnitt Standard-Cache-Verhalten unter Viewer-Protokollrichtlinie die Option Umleiten HTTP zu ausHTTPS.

Wenn Sie diese Funktion auswählen, werden HTTP Anfragen automatisch weitergeleitet, HTTPS um Ihre Website und die Daten Ihrer Zuschauer zu schützen.

9. Behalten Sie für die anderen Einstellungen im Abschnitt Standard-Cacheverhalten die Standardwerte bei.

(Optional) Sie können steuern, wie lange Ihre Datei in einem CloudFront Cache CloudFront verbleibt, bevor eine weitere Anfrage an Ihren Ursprung weitergeleitet wird. Eine Reduzierung der Dauer ermöglicht Ihnen, dynamische Inhalte bereitzustellen. Eine Erhöhung der Dauer bedeutet, dass Ihre Viewer eine bessere Leistung erhalten, da es wahrscheinlicher ist, dass Ihre Dateien direkt vom Edge-Cache bereitgestellt werden. Eine längere Dauer verringert darüber hinaus die Last auf Ihrem Ursprung. Weitere Informationen finden Sie unter [Verwaltung der Verweildauer von Inhalten im Cache \(Ablauf\)](#) im Amazon CloudFront Developer Guide.

10. Behalten Sie für die anderen Abschnitte die restlichen Einstellungen auf die Standardwerte eingestellt.

Weitere Informationen zu den verschiedenen Einstellungsoptionen finden Sie unter [Werte, die Sie beim Erstellen oder Aktualisieren einer Distribution angeben](#) im Amazon CloudFront Developer Guide.

11. Klicken Sie unten auf der Seite auf Verteilung erstellen.
12. Auf der Registerkarte Allgemein für Ihre CloudFront Distribution ändert sich unter Details der Wert der Spalte Letzte Änderung für Ihre Distribution von Bereitstellen auf den Zeitstempel, zu dem die Verteilung zuletzt geändert wurde. Dieser Prozess dauert in der Regel einige Minuten.

Überprüfen der Bucket-Richtlinie

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Buckets-Liste den Namen des Buckets aus, den Sie zuvor als Ursprung Ihrer CloudFront Distribution verwendet haben (z. B. **tutorial-bucket**).
4. Wählen Sie die Registerkarte Permissions (Berechtigungen).
5. Bestätigen Sie im Abschnitt Bucket-Richtlinie, dass eine Anweisung ähnlich der folgenden im Bucket-Richtlinientext angezeigt wird:

```
{
  "Version": "2008-10-17",
  "Id": "PolicyForCloudFrontPrivateContent",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
Identity EH1HDMB1FH2TC"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::tutorial-bucket/*"
    }
  ]
}
```

Dies ist die Aussage, die Ihre CloudFront Distribution zu Ihrer Bucket-Richtlinie hinzugefügt hat, als Sie Ja, die Bucket-Richtlinie früher aktualisieren ausgewählt haben.

Diese Aktualisierung der Bucket-Richtlinie weist darauf hin, dass Sie die CloudFront Distribution erfolgreich konfiguriert haben, um den Zugriff auf den S3-Bucket einzuschränken. Aufgrund dieser Einschränkung kann auf Objekte im Bucket nur über Ihre CloudFront Distribution zugegriffen werden.

Schritt 5: Greifen Sie über die CloudFront Distribution auf das Video zu

Jetzt CloudFront können Sie das in Ihrem S3-Bucket gespeicherte Video bereitstellen. Um über auf Ihr Video zugreifen zu können CloudFront, müssen Sie Ihren CloudFront Vertriebsdomännennamen mit dem Pfad zum Video im S3-Bucket kombinieren.

Um ein Video zum S3 URL zu erstellen, verwenden Sie den Namen der CloudFront Distributionsdomain

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Um den Namen der Verteilungsdomäne zu erhalten, gehen Sie wie folgt vor:
 - a. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach dem Ursprungsnamen suchen, der mit dem S3-Bucket beginnt, den Sie in [Schritt 1](#) erstellt haben (z. B. **tutorial-bucket**).
 - b. Nachdem Sie die Verteilung in der Liste gefunden haben, erweitern Sie die Spalte Domainname, um den Domainnamenwert für Ihre CloudFront Distribution zu kopieren.
4. Fügen Sie in einer neuen Browser-Registerkarte den kopierten Verteilungsdomännennamen ein.
5. Kehren Sie zum vorherigen Browser-Tab zurück und öffnen Sie die S3-Konsole unter <https://console.aws.amazon.com/s3/>.
6. Wählen Sie im linken Navigationsbereich Buckets aus.
7. Wählen Sie in der Liste der Buckets den Namen des Buckets, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben.
8. Wählen Sie in der Liste Objekte den Namen des Videos aus, das Sie in [Schritt 2](#) (z. B. `sample.mp4`) hochgeladen haben.
9. Kopieren Sie auf der Objektdetailseite im Übersicht über das Objekt den Wert des Schlüssel. Dieser Wert ist der Pfad zum hochgeladenen Videoobjekt im S3-Bucket.
10. Kehren Sie zur Browser-Registerkarte zurück, in die Sie zuvor den Namen der Verteilungsdomäne eingefügt haben und geben Sie einen Schrägstrich (/) nach dem Domainnamen der Distributionsdomäne ein und fügen Sie dann den Pfad zu dem Video, das Sie vorher kopiert haben, ein (zum Beispiel `sample.mp4`).

Jetzt ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront einen Server gehostetURL, der wie folgt aussieht:

```
https://CloudFront distribution domain name/Path to the S3 video
```

Ersetzen *CloudFront distribution domain name* and *Path to the S3 video* mit den entsprechenden Werten. Ein Beispiel einer URL ist **https://d111111abcdef8.cloudfront.net/sample.mp4**.

Schritt 6: Konfigurieren Sie Ihre CloudFront Distribution so, dass sie Ihren benutzerdefinierten Domainnamen verwendet

Um Ihren eigenen Domainnamen anstelle des CloudFront Domainnamens zu verwenden, URL um auf das S3-Video zuzugreifen, fügen Sie Ihrer CloudFront Distribution einen alternativen Domainnamen hinzu.

Teilschritte

- [Fordern Sie ein SSL Zertifikat an](#)
- [Fügen Sie den alternativen Domainnamen zu Ihrer CloudFront Distribution hinzu](#)
- [Erstellen Sie einen DNS Eintrag, um den Traffic von Ihrem alternativen Domainnamen zum Domainnamen Ihrer CloudFront Distribution weiterzuleiten](#)
- [Prüfen Sie, ob diese Option für Ihren Vertrieb aktiviert IPv6 ist, und erstellen Sie bei Bedarf einen weiteren DNS Datensatz](#)

Fordern Sie ein SSL Zertifikat an

Um es Ihren Zuschauern zu ermöglichen, Ihren benutzerdefinierten Domainnamen URL für Ihr Videostreaming zu verwenden, verwenden Sie HTTPS AWS Certificate Manager (ACM), um ein Secure Sockets Layer (SSL) -Zertifikat anzufordern. Das SSL Zertifikat stellt eine verschlüsselte Netzwerkverbindung zur Website her.

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die ACM Konsole unter <https://console.aws.amazon.com/acm/>.
2. Wenn die Einführungsseite angezeigt wird, klicken Sie unter Bereitstellung von Zertifikaten auf Erste Schritte.
3. Wählen Sie auf der Seite Zertifikat anfordern die Option Öffentliches Zertifikat anfordern und dann Zertifikat anfordern aus.

4. Geben Sie auf der Seite Domainnamen hinzufügen den vollqualifizierten Domainnamen (FQDN) der Site ein, die Sie mit einem SSL TLS /-Zertifikat sichern möchten. Sie können ein Sternchen (*) verwenden, um ein Platzhalterzertifikat anzufordern, um mehrere Webseitenamen in derselben Domäne zu schützen. Geben Sie in diesem Tutorial * und den benutzerdefinierten Domännennamen ein, den Sie in [Voraussetzungen](#) konfiguriert haben. Verwenden Sie zum Beispiel *.**example.com** und wählen Sie dann Weiter aus.

Weitere Informationen finden Sie unter [So fordern Sie ein ACM öffentliches Zertifikat an \(Konsole\)](#) im AWS Certificate Manager Benutzerleitfaden.

5. Wählen Sie auf der Seite „Validierungsmethode auswählen“ die Option DNSValidierung aus. Wählen Sie anschließend Weiter.

Wenn Sie Ihre DNS Konfiguration bearbeiten können, empfehlen wir Ihnen, die DNS Domain-Validierung anstelle der E-Mail-Validierung zu verwenden. DNSDie Validierung hat mehrere Vorteile gegenüber der E-Mail-Validierung. Weitere Informationen finden Sie unter [Option 1: DNS Validierung](#) in der AWS Certificate Manager Benutzerleitfaden.

6. (Optional) Auf der Seite Tags hinzufügen können Sie Ihr Zertifikat mit Metadaten markieren.
7. Wählen Sie Überprüfen aus.
8. Überprüfen Sie auf der Seite Prüfen, ob die Informationen unter Domänenname und Validierungsmethode korrekt sind. Wählen Sie Bestätigen und anfordern aus.

Auf der Validierungs-Seite wird angezeigt, dass Ihre Anforderung verarbeitet wird und dass Zertifikatdomänen validiert werden. Das Zertifikat, das auf die Validierung wartet, befindet sich im Status Validierung ausstehend.

9. Wählen Sie auf der Seite Validierung den Abwärtspfeil links neben Ihrem benutzerdefinierten Domainnamen und wählen Sie dann Create record in Route 53 aus, um Ihren Domainbesitz zu validierenDNS.

Dadurch wird ein CNAME Datensatz hinzugefügt, der bereitgestellt wird von AWS Certificate Manager zu Ihrer DNS Konfiguration.

10. Wählen Sie im Dialogfeld Datensatz in Route 53 erstellen die Option Erstellen.

Die Seite Validierung sollte unten eine Statusbenachrichtigung von Erfolg anzeigen.

11. Wählen Sie Continue (Fortfahren), um die Listenseite Certificates (Zertifikate) anzuzeigen.

Der Status für Ihr neues Zertifikat wird sich innerhalb von 30 Minuten von Ausstehende Validierung auf Ausgestellt ändern.

Fügen Sie den alternativen Domainnamen zu Ihrer CloudFront Distribution hinzu

1. Melden Sie sich bei AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Wählen Sie die ID für die Verteilung aus, die Sie in [Schritt 4](#) erstellt haben.
4. Gehen Sie auf der Registerkarte Allgemeines zum Abschnitt Einstellungen und wählen Sie Bearbeiten aus.
5. Wählen Sie auf der Seite Einstellungen bearbeiten für Alternativer Domainname (CNAME) — optional die Option Element hinzufügen aus, um die benutzerdefinierten Domainnamen hinzuzufügen, die Sie in dem URL von dieser CloudFront Distribution bereitgestellten S3-Video verwenden möchten.

Wenn Sie in diesem Tutorial beispielsweise Datenverkehr für eine Subdomäne weiterleiten möchten, z. B. `www.example.com`, geben Sie den Subdomännennamen (`www`) mit dem Domännennamen (`example.com`) ein. Geben Sie insbesondere **`www.example.com`** ein.

Note

Der alternative Domainname (CNAME), den Sie hinzufügen, muss durch das SSL Zertifikat abgedeckt sein, das Sie zuvor Ihrer CloudFront Distribution beigefügt haben.

6. Wählen Sie unter Benutzerdefiniertes SSL Zertifikat — optional das SSL Zertifikat aus, das Sie zuvor angefordert haben (z. B. **`*.example.com`**).

Note

Wenn das SSL Zertifikat nicht sofort nach der Anforderung angezeigt wird, warten Sie 30 Minuten, und aktualisieren Sie dann die Liste, bis das SSL Zertifikat für Sie zur Auswahl verfügbar ist.

7. Lassen Sie die restlichen Einstellungen bei den Standardwerten. Wählen Sie Save Changes (Änderungen speichern).
8. Warten Sie auf der Registerkarte Allgemeines für die Verteilung, bis der Wert Zuletzt geändert von Wird bereitgestellt in den Zeitstempel geändert wurde, als die Verteilung zuletzt geändert wurde.

Erstellen Sie einen DNS Eintrag, um den Traffic von Ihrem alternativen Domainnamen zum Domainnamen Ihrer CloudFront Distribution weiterzuleiten

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.
2. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).
3. Wählen Sie auf der Seite Gehostete Zonen den Namen der gehosteten Zone aus, die Route 53 für Sie in [Voraussetzungen](#) erstellt hat (z. B. **example.com**).
4. Klicken Sie auf Datensatz erstellen und benutzen Sie dann die Datensatz Quick Create-Methode.
5. Behalten Sie für den Datensatznamen den Wert für den Datensatznamen bei, den CloudFront Sie zuvor hinzugefügt haben.

Geben Sie in diesem Tutorial den Namen der Subdomäne ohne den Domänennamen ein, um Datenverkehr an eine Subdomäne weiterzuleiten, z. B. `www.example.com`. Geben Sie z. B. nur ein **www** im Textfeld vor Ihrem benutzerdefinierten Domänennamen ein.

6. Wählen Sie als Datensatztyp die Option A — Leitet den Verkehr an eine IPv4 Adresse weiter und einige AWS Ressourcen.
7. Wählen Sie für Wert den Alias-Toggle, um die Alias-Ressource zu aktivieren.
8. Wählen Sie unter Traffic weiterleiten an die Option Alias to CloudFront Distribution aus der Dropdownliste aus.
9. Wählen Sie im Suchfeld mit der Aufschrift Vertrieb auswählen den Domainnamen der CloudFront Distribution aus, die Sie in [Schritt 4](#) erstellt haben.

Gehen Sie wie folgt vor, um den Domainnamen Ihrer CloudFront Distribution zu finden:

- a. Melden Sie sich in einem neuen Browser-Tab bei AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v3/home>.
 - b. Wählen Sie im linken Navigationsbereich Verteilungen aus.
 - c. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach dem Ursprungsnamen suchen, der mit dem S3-Bucket beginnt, den Sie in [Schritt 1](#) erstellt haben (z. B. **tutorial-bucket**).
 - d. Nachdem Sie die Verteilung in der Liste gefunden haben, erweitern Sie die Spalte Domainname, um den Domainnamenwert für Ihre CloudFront Distribution zu sehen.
10. Behalten Sie auf der Seite Datensatz erstellen in der Route 53-Konsole die Standardwerte für die Übrigen Einstellungen bei.

11. Wählen Sie Create records (Datensätze erstellen).

Prüfen Sie, ob diese Option für Ihren Vertrieb aktiviert IPv6 ist, und erstellen Sie bei Bedarf einen weiteren DNS Datensatz

Wenn für Ihre Distribution aktiviert IPv6 ist, müssen Sie einen weiteren DNS Datensatz erstellen.

1. Gehen Sie wie folgt vor, um zu überprüfen, ob es für Ihre Distribution aktiviert IPv6 ist:
 - a. Melden Sie sich an bei AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
 - b. Wählen Sie im linken Navigationsbereich Verteilungen aus.
 - c. Wählen Sie die ID der CloudFront Distribution aus, die Sie in [Schritt 4](#) erstellt haben.
 - d. Überprüfen Sie auf der Registerkarte Allgemein unter Einstellungen, ob die Option Aktiviert ausgewählt IPv6 ist.

Wenn für Ihre Distribution aktiviert IPv6 ist, müssen Sie einen weiteren DNS Datensatz erstellen.

2. Wenn für Ihre Distribution aktiviert IPv6 ist, gehen Sie wie folgt vor, um einen DNS Datensatz zu erstellen:
 - a. Melden Sie sich an bei AWS Management Console und öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.
 - b. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).
 - c. Wählen Sie auf der Seite Gehostete Zonen den Namen der gehosteten Zone aus, die Route 53 für Sie in [Voraussetzungen](#) erstellt hat (z. B. **example.com**).
 - d. Klicken Sie auf Datensatz erstellen und benutzen Sie dann die Datensatz Quick Create-Methode.
 - e. Geben Sie für Datensatzname in das Textfeld vor Ihrem benutzerdefinierten Domainnamen denselben Wert ein, den Sie zuvor bei der Erstellung des IPv4 DNS Datensatzes eingegeben haben. Geben Sie in diesem Tutorial beispielsweise nur **www** ein, um den Datenverkehr für die Subdomäne `www.example.com` weiterzuleiten.
 - f. Wählen Sie als Datensatztyp die Option AAAA - Leitet den Verkehr an eine IPv6 Adresse weiter und einige AWS Ressourcen.
 - g. Wählen Sie für Wert den Alias-Toggle, um die Alias-Ressource zu aktivieren.

- h. Wählen Sie unter Traffic weiterleiten an die Option Alias to CloudFront Distribution aus der Dropdownliste aus.
- i. Wählen Sie im Suchfeld mit der Aufschrift Vertrieb auswählen den Domainnamen der CloudFront Distribution aus, die Sie in [Schritt 4](#) erstellt haben.
- j. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.
- k. Wählen Sie Create records (Datensätze erstellen).

Schritt 7: Greifen Sie über die CloudFront Distribution mit dem benutzerdefinierten Domainnamen auf das S3-Video zu

Um mithilfe der benutzerdefinierten Version auf das S3-Video zuzugreifen URL, müssen Sie Ihren alternativen Domainnamen mit dem Pfad zum Video im S3-Bucket kombinieren.

Um eine benutzerdefinierte Option für den URL Zugriff auf das S3-Video über die CloudFront Distribution zu erstellen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Gehen Sie wie folgt vor, um den alternativen Domainnamen Ihrer CloudFront Distribution zu erhalten:
 - a. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach ihrem Ursprungsnamen suchen. Dieser beginnt mit dem S3-Bucket-Namen für den Bucket, den Sie in [Schritt 1](#) erstellt haben (z. B. **tutorial-bucket**).
 - b. Nachdem Sie die Verteilung in der Liste gefunden haben, erweitern Sie die Spalte Alternative Domainnamen, um den Wert des alternativen Domainnamens Ihrer CloudFront Distribution zu kopieren.
4. Fügen Sie in einem neuen Browser-Tab den alternativen Domainnamen der CloudFront Distribution ein.
5. Kehren Sie zum vorherigen Browser-Tab zurück und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
6. Finden Sie den Pfad zu Ihrem S3-Video, wie im [Schritt 5](#) erklärt wird.

7. Kehren Sie zur Browserregisterkarte zurück, auf der Sie zuvor den alternativen Domännennamen eingefügt haben, geben Sie einen Vorwärts-Schrägstrich (/) ein und fügen Sie dann den Pfad in Ihr S3-Video ein (z. B. `sample.mp4`).

Jetzt ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront eine benutzerdefinierte Version gehostetURL, die wie folgt aussieht:

```
https://CloudFront distribution alternate domain name/Path to the S3 video
```

Ersetzen *CloudFront distribution alternate domain name* and *Path to the S3 video* mit den entsprechenden Werten. Ein Beispiel einer URList **`https://www.example.com/sample.mp4`**.

(Optional) Schritt 8: Zeigen Sie Daten zu Anfragen an, die bei Ihrer CloudFront Distribution eingegangen sind

Um Daten zu Anfragen anzuzeigen, die bei Ihrer CloudFront Distribution eingegangen sind

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie im linken Navigationsbereich unter Berichte & Analytik die Berichte aus der Konsole aus, die sich über Cache-Statistiken, Beliebte Objekte, Top-Referrer, Verwendung bis zu Viewer erstrecken.

Sie können jedes Berichts-Dashboard filtern. Weitere Informationen finden Sie unter [CloudFront Berichte in der Konsole](#) im Amazon CloudFront Developer Guide.

3. Um Daten zu filtern, wählen Sie die ID der CloudFront Distribution aus, die Sie in [Schritt 4](#) erstellt haben.

Schritt 9: Bereinigen

Wenn Sie ein S3-Streaming-Video mit CloudFront Route 53 nur als Lernübung gehostet haben, löschen Sie das AWS Ressourcen, die Sie so zugewiesen haben, dass Ihnen keine Gebühren mehr anfallen.

Note

Wenn Sie eine Domäne registrieren, kostet diese sofort Geld. Dies kann nicht rückgängig gemacht werden. Sie können wählen, die Domäne nicht automatisch zu erneuern, aber Sie bezahlen im Voraus und besitzen sie für das Jahr. Weitere Informationen finden Sie unter [Registrieren einer neuen Domäne](#) im Amazon-Route-53-Entwicklerhandbuch.

Teilschritte

- [Löschen Sie die Verteilung CloudFront](#)
- [Löschen Sie den Datensatz DNS](#)
- [Löschen Sie die öffentlich gehostete Zone für Ihre benutzerdefinierte Domäne](#)
- [Löschen des benutzerdefinierten Domännennamens aus Route 53](#)
- [Löschen des Originalvideos im S3-Quell-Bucket](#)
- [Löschen des S3-Quell-Bucket](#)

Löschen Sie die Verteilung CloudFront

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach ihrem Ursprungsnamen suchen. Dieser beginnt mit dem S3-Bucket-Namen für den Bucket, den Sie in [Schritt 1](#) erstellt haben (z. B. **tutorial-bucket**).
4. Um die CloudFront Distribution zu löschen, müssen Sie sie zuerst deaktivieren.
 - Wenn der Wert der Spalte Status Aktiviert ist und der Wert von Letzte Änderung der Zeitstempel ist, in dem die Verteilung zuletzt geändert wurde, deaktivieren Sie die Verteilung weiterhin, bevor Sie sie löschen.
 - Wenn der Wert von Status Aktiviert ist und der Wert von Letzte Änderung Wird bereitgestellt ist, warten Sie, bis sich der Wert von Status in den Zeitstempel ändert, als die Verteilung zuletzt geändert wurde. Fahren Sie dann fort, um die Verteilung vor dem Löschen zu deaktivieren.
5. Gehen Sie wie folgt vor, um die CloudFront Verteilung zu deaktivieren:

- a. Wählen Sie in der Verteilungen-Liste das Kontrollkästchen neben der ID für die Verteilung, die Sie löschen möchten.
- b. Um die Verteilung zu deaktivieren, wählen Sie Deaktivieren, und wählen Sie dann Deaktivieren, um zu bestätigen.

Wenn Sie eine Verteilung deaktivieren, der ein alternativer Domänenname zugeordnet ist, wird kein Verkehr CloudFront mehr für diesen Domainnamen (z. B. `www.example.com`) akzeptiert, auch wenn eine andere Distribution einen alternativen Domainnamen mit einem Platzhalter (*) hat, der derselben Domäne entspricht (z. B. `*.example.com`).

- c. Der Wert von State (Zustand) wird sofort in Disabled (Deaktiviert) geändert. Warten Sie bis der Wert Zuletzt geändert von Wird bereitgestellt in den Zeitstempel geändert wurde, als die Verteilung zuletzt geändert wurde.


Da diese Änderung auf alle Edge-Standorte übertragen werden CloudFront muss, kann es einige Minuten dauern, bis die Aktualisierung abgeschlossen ist und Sie mit der Option Löschen die Verteilung löschen können.

6. Gehen Sie wie folgt vor, um die deaktivierte Verteilung zu löschen:
 - a. Aktivieren Sie das Kontrollkästchen neben der ID für die Verteilung, die Sie löschen möchten.
 - b. Wählen Sie Löschen und Löschen aus, um den Vorgang zu bestätigen.

Löschen Sie den Datensatz DNS

Informationen zum Löschen der öffentlich gehosteten Zone für die Domain (einschließlich des DNS Datensatzes) finden Sie [Löschen Sie die öffentlich gehostete Zone für Ihre benutzerdefinierte Domäne](#) im Amazon Route 53 Developer Guide. Wenn Sie nur den in [Schritt 6](#) erstellten DNS Datensatz löschen möchten, gehen Sie wie folgt vor:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.
2. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).
3. Wählen Sie auf der Seite Gehostete Zonen den Namen der gehosteten Zone aus, die Route 53 für Sie in [Voraussetzungen](#) erstellt hat (z. B. `example.com`).
4. Wählen Sie in der Liste der Datensätze das Kontrollkästchen neben den Datensätzen aus, die Sie löschen möchten (die Datensätze, die Sie in [Schritt 6](#) erstellt haben).


 Note

Sie können keine Datensätze löschen, deren Type den Wert NS oder hat SOA.

5. Wählen Sie Delete Records (Datensätze löschen).
6. Um die Löschung zu bestätigen, klicken Sie auf Delete (Löschen).

Es dauert einige Zeit, bis Änderungen an Datensätzen auf die Route DNS 53-Server übertragen werden. [Derzeit besteht die einzige Möglichkeit, um zu überprüfen, ob Ihre Änderungen übernommen wurden, darin, die GetChange API Aktion zu verwenden.](#) Änderungen werden im Allgemeinen innerhalb von 60 Sekunden an alle Route 53-Name-Server übertragen.

Löschen Sie die öffentlich gehostete Zone für Ihre benutzerdefinierte Domäne

 Warning

Wenn Sie Ihre Domänenregistrierung behalten möchten, aber die Weiterleitung von Internetdatenverkehr an Ihre Website oder Webanwendung beenden möchten, empfehlen wir, dass Sie Datensätze in der gehosteten Zone (wie im obigen Abschnitt beschrieben) löschen, statt die gehostete Zone zu löschen.


Wenn Sie eine gehostete Zone löschen, könnte jemand anderes die Domäne nutzen und den Datenverkehr über Ihren Domännennamen an die eigenen Ressourcen weiterleiten.

Wenn Sie eine gehostete Zone löschen, können Sie diese außerdem nicht wiederherstellen. Sie müssen eine neue gehostete Zone erstellen und Sie die Namensserver für Ihre Domain-Registrierung aktualisieren. Es kann bis zu 48 Stunden dauern, bis diese Einstellungen wirksam werden.

Wenn Sie möchten, dass die Domain im Internet nicht verfügbar ist, können Sie Ihren DNS Dienst zunächst auf einen kostenlosen DNS Dienst übertragen und dann die von Route 53 gehostete Zone löschen. Dadurch wird verhindert, dass future DNS Anfragen möglicherweise falsch weitergeleitet werden.

1. Wenn die Domain bei Route 53 registriert ist, finden Sie unter [Hinzufügen oder Ändern von Nameservern und Glue-Datensätzen für eine Domain](#) im Amazon Route 53-Entwicklerhandbuch Informationen darüber, wie Sie Route 53-Nameserver durch Nameserver für den neuen DNS Service ersetzen können.

2. Wenn die Domäne mit einer anderen Vergabestelle registriert wurde, verwenden Sie die Methode der Vergabestelle, um Namensserver für die Domäne zu ändern.

 Note

Wenn Sie eine gehostete Zone für eine Subdomäne (`www.example.com`) löschen, müssen Sie keine Namensserver für die Domäne (`example.com`) ändern.

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.
2. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).
3. Wählen Sie auf der Seite Hosted Zones den Namen für die gehostete Zone aus, die Sie löschen möchten.
4. Vergewissern Sie sich auf der Registerkarte Datensätze für Ihre gehostete Zone, dass die Hosting-Zone, die Sie löschen möchten, nur einen NS und einen SOA Datensatz enthält.

Enthält sie zusätzliche Datensätze, löschen Sie diese zuerst.

Wenn Sie NS-Datensätze für Subdomänen in der gehosteten Zone erstellt haben, löschen Sie auch diese Datensätze.

5. Deaktivieren Sie auf der Registerkarte DNSSEC Signatur für Ihre gehostete Zone das DNNSSEC Signieren, falls es aktiviert war. Weitere Informationen finden Sie unter [Deaktivieren der DNSSEC Signierung](#) im Amazon Route 53-Entwicklerhandbuch.
6. Wählen Sie oben auf der Detailseite der gehosteten Zone Zone löschen aus.
7. Geben Sie **delete** ein, um das Löschen zu bestätigen, und wählen Sie dann Löschen.

Löschen des benutzerdefinierten Domänennamens aus Route 53

Bei den meisten Top-Level-Domains (TLDs) können Sie die Registrierung löschen, wenn Sie sie nicht mehr benötigen. Wenn Sie eine Domainnamenregistrierung von Route 53 löschen, bevor die Registrierung abläuft, AWS erstattet die Registrierungsgebühr nicht. Weitere Informationen finden Sie unter [Löschen einer Domainnamenregistrierung](#) im Amazon-Route 53-Entwicklerhandbuch.

⚠ Important

Wenn Sie die Domain übertragen möchten zwischen AWS-Konten oder übertragen Sie die Domain an einen anderen Registrar, löschen Sie die Domain nicht und gehen Sie davon aus, dass Sie sie sofort erneut registrieren werden. Lesen Sie stattdessen die entsprechende Dokumentation im Entwicklerhandbuch für Amazon Route 53:

- [Übertragung einer Domain auf eine andere AWS-Konto](#)
- [Überträgt eine Domain von Amazon Route 53 zu einer anderen Vergabestelle](#)

Löschen des Originalvideos im S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Bucket-Name den Namen des Buckets aus, in den Sie das Video in [Schritt 2](#) (z. B. **tutorial-bucket**) hochgeladen haben.
4. Markieren Sie auf der Registerkarte Objekte das Kontrollkästchen neben dem Namen des Objekts, das Sie löschen möchten (z. B. `sample.mp4`).
5. Wählen Sie Löschen.
6. Geben Sie unter Objekte endgültig löschen? den Wert **permanently delete** um zu bestätigen, dass Sie dieses Objekt löschen möchten.
7. Wählen Sie Delete objects (Objekte löschen).

Löschen des S3-Quell-Bucket

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie in [Schritt 1](#) (z. B. **tutorial-bucket**) erstellt haben.
4. Wählen Sie Löschen.

5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Nächste Schritte

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie die folgenden verwandten Anwendungsfälle näher untersuchen:

- Transkodieren Sie S3-Videos in Streaming-Formate, die für ein bestimmtes Fernsehgerät oder ein angeschlossenes Gerät benötigt werden, bevor Sie diese Videos bei einer CloudFront Distribution hosten.

Um Amazon S3 Batch Operations zu verwenden, AWS Lambda and AWS Elemental MediaConvert Informationen zur Batch-Transcodierung einer Sammlung von Videos in eine Vielzahl von Ausgabemedienformaten finden Sie unter [Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations, AWS Lambda, und AWS Elemental MediaConvert](#)

- Hosten Sie andere in S3 gespeicherte Objekte wie Bilder, Audio, Animationsgrafiken, StylesheetsHTML, JavaScript,, React-Apps usw. mithilfe von Route CloudFront 53.

Ein Beispiel finden Sie unter [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#) und [Beschleunigen Sie Ihre Website mit Amazon CloudFront](#).

- Verwenden Sie [Amazon S3 Transfer Acceleration](#), um schnelle und sichere Dateiübertragungen zu konfigurieren. Transfer Acceleration kann das Hochladen von Videos zu Ihrem S3-Bucket zur Übertragung größerer Videos über große Entfernungen beschleunigen. Transfer Acceleration verbessert die Übertragungsleistung, indem der Datenverkehr über die CloudFront weltweit verteilten Edge-Standorte und über AWS Backbone-Netzwerke. Es verwendet auch Netzwerkprotokoll-Optimierungen. Weitere Informationen finden Sie unter [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#).

Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations, AWS Lambda, und AWS Elemental MediaConvert

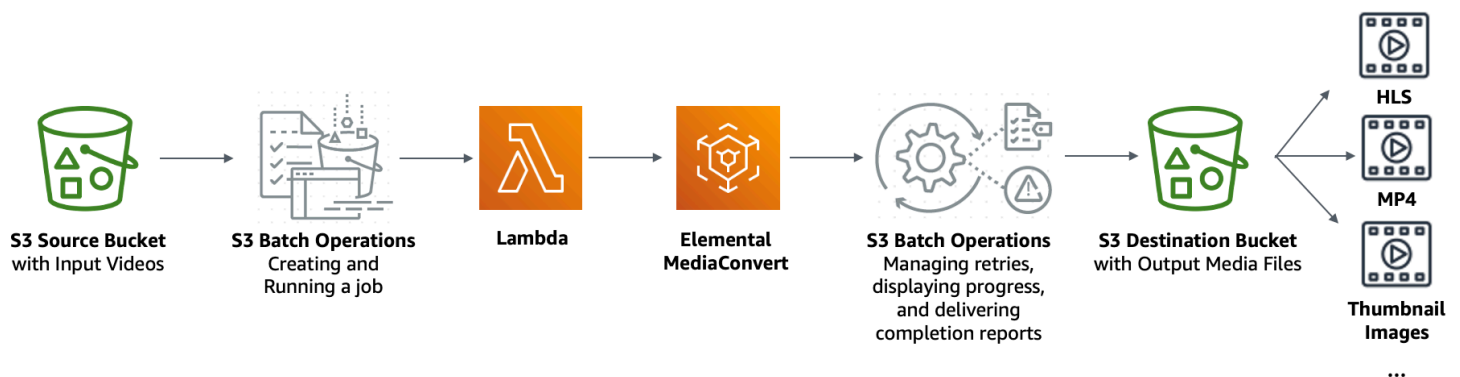
Videokonsumenten nutzen Geräte aller Formen, Größen und Jahrgänge, um Medieninhalte zu genießen. Diese breite Palette an Geräten stellt eine Herausforderung für Content-Ersteller und -

Distributoren dar. Videos müssen nicht in einem one-size-fits-all Format vorliegen, sondern müssen so konvertiert werden, dass sie ein breites Spektrum an Größen, Formaten und Bitraten abdecken können. Diese Konvertierungsaufgabe ist noch schwieriger, wenn Sie eine große Anzahl von Videos haben, die konvertiert werden müssen.

AWS bietet Ihnen eine Methode zum Aufbau einer skalierbaren, verteilten Architektur mit folgenden Funktionen:

- Greift Eingabevideos auf
- Verarbeitet die Videos zur Wiedergabe auf einer Vielzahl von Geräten
- Speichert die transcodierten Mediendateien
- Liefert die Ausgabemediendateien und befriedigt Ansprüche

Wenn Sie umfangreiche Video-Repositories in Amazon S3 gespeichert haben, können Sie diese Videos aus ihren Quellformaten in mehrere Dateitypen in der Größe, Auflösung und Format transkodieren, die von einem bestimmten Videoplayer oder Gerät benötigt werden. Insbesondere bietet Ihnen [S3 Batch Operations](#) eine Lösung zum Aufrufen AWS Lambda Funktionen für bestehende Eingabevideos in einem S3-Quell-Bucket. Dann rufen die Lambda-Funktionen auf [AWS Elemental MediaConvert](#) umfangreiche Videotranskodierungsaufgaben durchzuführen. Die konvertierten Ausgabe-Mediendateien werden in einem S3-Ziel-Bucket gespeichert.



Ziel

In diesem Lernprogramm erfahren Sie, wie Sie S3-Batch-Vorgänge einrichten, um eine Lambda-Funktion zum Batch-Transcodieren von Videos aufzurufen, die in einem S3-Quell-Bucket gespeichert sind. Die Lambda-Funktion ruft MediaConvert auf, um die Videos zu transkodieren. Die Ausgänge für jedes Video im S3-Quell-Bucket sind wie folgt:

- Ein [HTTPLive-Streaming-Stream \(HLS\)](#) mit adaptiver Bitrate für die Wiedergabe auf Geräten verschiedener Größen und unterschiedlicher Bandbreiten
- Eine Videodatei MP4
- In Intervallen gesammelte Miniaturbilder

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie einen S3-Bucket für Ausgabe-Mediendateien](#)
- [Schritt 2: Erstelle eine IAM Rolle für MediaConvert](#)
- [Schritt 3: Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion](#)
- [Schritt 4: Erstellen einer Lambda-Funktion für die Videotranscodierung](#)
- [Schritt 5: Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket](#)
- [Schritt 6: Eine IAM Rolle für S3 Batch Operations erstellen](#)
- [Schritt 7: Einrichten und Ausführen eines Auftrags für S3-Batchvorgänge](#)
- [Schritt 8: Überprüfen Sie die Ausgabe-Mediendateien aus Ihrem S3-Ziel-Bucket](#)
- [Schritt 9: Bereinigen](#)
- [Nächste Schritte](#)

Voraussetzungen

Bevor Sie mit diesem Lernprogramm beginnen können, benötigen Sie einen Amazon S3-Quell-Bucket (z. B. **tutorial-bucket-1**) mit Videos, die bereits darin transcodiert werden sollen.

Sie können dem Bucket einen anderen Namen geben, wenn Sie möchten. Weitere Informationen zu Amazon S3 Bucket-Namen finden Sie unter [Regeln für die Benennung von Buckets](#).

Lassen Sie für den S3-Quell-Bucket die Einstellungen für öffentlichen Zugriff für diesen Bucket blockieren auf die Standardeinstellungen festlegen (Alle öffentlichen Zugriffe blockieren ist aktiviert). Weitere Informationen finden Sie unter [Erstellen eines Buckets](#).

Weitere Informationen über das Hochladen von Videos in den S3-Quell-Bucket finden Sie unter [Objekte hochladen](#). Wenn Sie ein Video in S3 hochladen, können Sie auch [Amazon S3 Transfer Acceleration](#) nutzen, um schnelle und sichere Dateiübertragungen zu konfigurieren. Transfer Acceleration kann das Hochladen von Videos zu Ihrem S3-Bucket zur Übertragung größerer Videos

über große Entfernungen beschleunigen. Weitere Informationen finden Sie unter [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#).

Schritt 1: Erstellen Sie einen S3-Bucket für Ausgabe-Mediendateien

In diesem Schritt erstellen Sie einen S3-Ziel-Bucket, um die konvertierten Ausgabe-Mediendateien zu speichern. Sie erstellen auch eine Cross-Origin-Konfiguration für die gemeinsame Nutzung von Ressourcen (CORS), um den ursprungsübergreifenden Zugriff auf die transkodierte Mediendateien zu ermöglichen, die in Ihrem S3-Ziel-Bucket gespeichert sind.

Teilschritte

- [Erstellen Sie einen Bucket für die Ausgabe-Mediendateien](#)
- [Fügen Sie dem S3-Ausgabe-Bucket eine CORS Konfiguration hinzu](#)

Erstellen Sie einen Bucket für die Ausgabe-Mediendateien

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.
4. Geben Sie im Feld Bucket Name einen Namen für Ihren Bucket ein (z. B. **tutorial-bucket-2**).
5. Wählen Sie für Region die AWS-Region wo sich der Bucket befinden soll.
6. Um den öffentlichen Zugriff auf Ihre Ausgabemediendateien zu gewährleisten, deaktivieren Sie unter Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren Alle öffentlichen Zugriffe blockieren.

Warning

Bevor Sie diesen Schritt ausführen, lesen Sie den Abschnitt [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#), um sicherzustellen, dass Sie die mit dem Zulassen eines öffentlichen Zugriffs verbundenen Risiken kennen und akzeptieren. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

Wenn Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs nicht löschen möchten, können Sie Amazon verwenden, CloudFront um die transkodierte Mediendateien an Zuschauer (Endbenutzer) zu senden. Weitere Informationen finden Sie unter [Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53](#).

7. Aktivieren Sie das Kontrollkästchen neben I acknowledge that the current settings may result in this bucket and the objects within becoming public (Ich bestätige, dass die aktuellen Einstellungen dazu führen können, dass dieser Bucket und die darin enthaltenen Objekte öffentlich werden).
8. Lassen Sie die restlichen Einstellungen bei den Standardwerten.
9. Wählen Sie Bucket erstellen aus.

Fügen Sie dem S3-Ausgabe-Bucket eine CORS Konfiguration hinzu

Eine JSON CORS Konfiguration definiert eine Methode für Client-Webanwendungen (in diesem Kontext Videoplayer), die in eine Domain geladen werden, transkodierte Ausgabemediendateien in einer anderen Domain abzuspielen.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste der Buckets den Namen des Buckets, den Sie zuvor erstellt haben (z. B. **tutorial-bucket-2**).
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Cross-Origin Resource Sharing (CORS) die Option Bearbeiten aus.
6. Kopieren Sie die folgende CORS Konfiguration und fügen Sie sie in das Textfeld für die CORS Konfiguration ein.

Die CORS Konfiguration muss im JSON Format vorliegen. In diesem Beispiel verwendet das AllowedOrigins-Attribut das Platzhalterzeichen (*) um alle Ursprünge anzugeben. Wenn du deine spezifische Herkunft kennst, kannst du das AllowedOrigins Attribut auf deinen spezifischen Spieler beschränkenURL. Weitere Informationen zum Konfigurieren von diesem und anderen Attributen finden Sie unter [Elemente einer CORS Konfiguration](#).

```
[
```

```
{
  "AllowedOrigins": [
    "*"
  ],
  "AllowedMethods": [
    "GET"
  ],
  "AllowedHeaders": [
    "*"
  ],
  "ExposeHeaders": []
}
```

7. Wählen Sie Änderungen speichern.

Schritt 2: Erstelle eine IAM Rolle für MediaConvert

Zur Verwendung AWS Elemental MediaConvert Um die in Ihrem S3-Bucket gespeicherten Eingabevideos zu transkodieren, benötigen Sie eine AWS Identity and Access Management (IAM) Servicerolle, um MediaConvert Berechtigungen zum Lesen und Schreiben von Videodateien aus und in Ihre S3-Quell- und Ziel-Buckets zu erteilen. Wenn Sie Transcodierungsaufträge ausführen, verwendet die MediaConvert Konsole diese Rolle.

Um eine IAM Rolle zu erstellen für MediaConvert

1. Erstellen Sie eine IAM Rolle mit einem Rollennamen, den Sie wählen (z. B. **tutorial-mediaconvert-role**). Um diese Rolle zu erstellen, folgen Sie den Schritten [unter Erstellen Sie Ihre MediaConvert Rolle in IAM \(Konsole\)](#) im AWS Elemental MediaConvert Benutzerleitfaden.
2. Nachdem Sie die IAM Rolle für erstellt haben MediaConvert, wählen Sie in der Liste der Rollen den Namen der Rolle aus MediaConvert , für die Sie erstellt haben (z. B. **tutorial-mediaconvert-role**).
3. Kopieren Sie auf der Übersichtsseite die Rolle ARN (die mit `arn:aws:iam::`) und speichern Sie sie ARN zur späteren Verwendung.

Weitere Informationen zu finden Sie ARNs unter [Amazon Resource Names \(ARNs\)](#) in der AWS Allgemeine Referenz.

Schritt 3: Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion

Um Videos mit MediaConvert und S3 Batch Operations stapelweise zu transkodieren, verwenden Sie eine Lambda-Funktion, um diese beiden Dienste zu verbinden, um Videos zu konvertieren. Diese Lambda-Funktion muss über eine IAM Rolle verfügen, die der Lambda-Funktion Zugriffsberechtigungen MediaConvert und S3-Stapeloperationen gewährt.

Teilschritte

- [Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion](#)
- [Betten Sie eine Inline-Richtlinie für die IAM Rolle Ihrer Lambda-Funktion ein](#)

Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen) und dann Create Role (Rolle erstellen) aus.
3. Wählen Sie das Symbol AWS Geben Sie die Servicerolle ein und wählen Sie dann unter Allgemeine Anwendungsfälle die Option Lambda aus.
4. Wählen Sie Weiter: Berechtigungen aus.
5. Geben Sie auf der Seite Attach permissions (Berechtigungen anfügen) **AWSLambdaBasicExecutionRole** in das Suchfeld ein, um die Liste der Richtlinien zu filtern. Um die verwaltete Richtlinie AWSLambdaBasicExecutionRole an diese Rolle anzuhängen, um Schreibberechtigungen für Amazon CloudWatch Logs zu gewähren, aktivieren Sie das Kontrollkästchen neben AWSLambdaBasicExecutionRole.
6. Wählen Sie Weiter: Markierungen.
7. (Optional) Fügen Sie der verwalteten Richtlinie Tags hinzu.
8. Wählen Sie Weiter: Prüfen aus.
9. Geben Sie für Role name (Rollenname) den Namen **tutorial-lambda-transcode-role** ein.
10. Wählen Sie Create role (Rolle erstellen) aus.

Betten Sie eine Inline-Richtlinie für die IAM Rolle Ihrer Lambda-Funktion ein

Um der MediaConvert Ressource, die für die Ausführung der Lambda-Funktion benötigt wird, Berechtigungen zu gewähren, müssen Sie eine Inline-Richtlinie verwenden.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Wählen Sie in der Liste Rollen den Namen der IAM Rolle aus, die Sie zuvor für Ihre Lambda-Funktion erstellt haben (z. B. **tutorial-lambda-transcode-role**).
4. Wählen Sie den Tab Permissions (Berechtigungen).
5. Wählen Sie Add inline Policy (Inline-Richtlinie auswählen).
6. Wählen Sie die JSONRegisterkarte aus, kopieren Sie dann die folgende JSON Richtlinie und fügen Sie sie ein.

Ersetzen Sie in der JSON Richtlinie den ARN Beispielwert von Resource durch die Rolle ARN der IAM Rolle MediaConvert , für die Sie in [Schritt 2](#) erstellt haben (z. B. **tutorial-mediaconvert-role**).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "Logging"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/tutorial-mediaconvert-role"
      ],
      "Effect": "Allow",
```

```
        "Sid": "PassRole"
    },
    {
        "Action": [
            "mediaconvert:*"
        ],
        "Resource": [
            "*"
        ],
        "Effect": "Allow",
        "Sid": "MediaConvertService"
    },
    {
        "Action": [
            "s3:*"
        ],
        "Resource": [
            "*"
        ],
        "Effect": "Allow",
        "Sid": "S3Service"
    }
]
}
```

7. Wählen Sie Review policy (Richtlinie überprüfen) aus.
8. Geben Sie unter Name **tutorial-lambda-policy** ein.
9. Wählen Sie Create Policy (Richtlinie erstellen) aus.

Nachdem Sie eine Inline-Richtlinie erstellt haben, wird sie automatisch in die IAM Rolle Ihrer Lambda-Funktion eingebettet.

Schritt 4: Erstellen einer Lambda-Funktion für die Videotranscodierung

In diesem Abschnitt des Tutorials erstellen Sie eine Lambda-Funktion mithilfe von SDK for Python zur Integration mit S3 Batch Operations und MediaConvert. Um mit der Transcodierung der Videos zu beginnen, die bereits in Ihrem S3-Quell-Bucket gespeichert sind, führen Sie einen S3-Batchvorgang-Auftrag aus, der direkt die Lambda-Funktion für jedes Video im S3-Quell-Bucket aufruft. Anschließend sendet die Lambda-Funktion einen Transcodierungsauftrag für jedes Video an MediaConvert

Teilschritte

- [Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets](#)
- [Erstellen einer Lambda-Funktion mit einer Ausführungsrolle \(Konsole\)](#)
- [Stellen Sie Ihre Lambda-Funktion mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion \(Konsole\)](#)

Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets

1. Erstellen Sie auf Ihrem lokalen Computer einen Ordner mit dem Namen `batch-transcode`.
2. Erstellen Sie im `batch-transcode` Ordner eine Datei mit JSON den Auftragseinstellungen. Sie können z. B. die Einstellungen in diesem Abschnitt verwenden und die Datei `job.json` benennen.

Eine `job.json`-Datei gibt Folgendes an:

- Welche Dateien transcodiert werden sollen
- Wie Sie Ihre Eingabevideos transcodieren möchten
- Welche Ausgabemediendateien Sie erstellen möchten
- Wie die transcodierten Dateien benannt werden sollen
- Wo die transcodierten Dateien gespeichert werden sollen
- Welche fortschrittlichen Funktionen angewendet werden sollen und so weiter

In diesem Tutorial wird die folgende `job.json`-Datei verwendet, um die folgenden Ausgaben für jedes Video im S3-Quell-Bucket zu erstellen:

- Ein HTTP Live-Streaming-Stream (HLS) mit adaptiver Bitrate für die Wiedergabe auf mehreren Geräten unterschiedlicher Größe und Bandbreite
- Eine Videodatei MP4
- In Intervallen gesammelte Miniaturbilder

Diese `job.json` Beispieldatei verwendet die qualitätsdefinierte variable Bitrate (QVBR), um die Videoqualität zu optimieren. Die HLS Ausgabe ist Apple-kompatibel (Audio ohne Mischung aus Video, Segmentdauer von 6 Sekunden und optimierte Videoqualität durch Auto). QVBR

Wenn Sie die hier bereitgestellten Beispieleinstellungen nicht verwenden möchten, können Sie eine `job.json`-Spezifikation basierend auf Ihrem Anwendungsfall erstellen. Stellen Sie sicher, dass die Eingabedateien ähnliche Video- und Audiokonfigurationen aufweisen, um die Konsistenz Ihrer Ausgänge zu gewährleisten. Für beliebige Eingabedateien mit unterschiedlichen Video- und Audiokonfigurationen erstellen Sie getrennte Automatisierungen (eindeutige `job.json`-Einstellungen). Weitere Informationen finden Sie unter Beispiel [AWS Elemental MediaConvert Jobeinstellungen JSON](#) in der AWS Elemental MediaConvert Benutzerleitfaden.

```
{
  "OutputGroups": [
    {
      "CustomName": "HLS",
      "Name": "Apple HLS",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "TimedMetadata": "NONE",
              "VideoPid": 481,
              "AudioPids": [
                482,
                483,
                484,
                485,
                486,
                487,
                488,
                489,
                490,
                491,
                492
              ]
            }
          }
        }
      ]
    }
  ]
}
```

```
},
"VideoDescription": {
  "Width": 640,
  "ScalingBehavior": "DEFAULT",
  "Height": 360,
  "TimecodeInsertion": "DISABLED",
  "AntiAlias": "ENABLED",
  "Sharpness": 50,
  "CodecSettings": {
    "Codec": "H_264",
    "H264Settings": {
      "InterlaceMode": "PROGRESSIVE",
      "NumberReferenceFrames": 3,
      "Syntax": "DEFAULT",
      "Softness": 0,
      "GopClosedCadence": 1,
      "GopSize": 2,
      "Slices": 1,
      "GopBReference": "DISABLED",
      "MaxBitrate": 1200000,
      "SlowPal": "DISABLED",
      "SpatialAdaptiveQuantization": "ENABLED",
      "TemporalAdaptiveQuantization": "ENABLED",
      "FlickerAdaptiveQuantization": "DISABLED",
      "EntropyEncoding": "CABAC",
      "FramerateControl": "INITIALIZE_FROM_SOURCE",
      "RateControlMode": "QVBR",
      "CodecProfile": "MAIN",
      "Telecine": "NONE",
      "MinIInterval": 0,
      "AdaptiveQuantization": "HIGH",
      "CodecLevel": "AUTO",
      "FieldEncoding": "PAFF",
      "SceneChangeDetect": "TRANSITION_DETECTION",
      "QualityTuningLevel": "SINGLE_PASS_HQ",
      "FramerateConversionAlgorithm": "DUPLICATE_DROP",
      "UnregisteredSeiTimecode": "DISABLED",
      "GopSizeUnits": "SECONDS",
      "ParControl": "INITIALIZE_FROM_SOURCE",
      "NumberBFramesBetweenReferenceFrames": 2,
      "RepeatPps": "DISABLED"
    }
  }
},
"AfdSignaling": "NONE",
```

```
    "DropFrameTimecode": "ENABLED",
    "RespondToAfd": "NONE",
    "ColorMetadata": "INSERT"
  },
  "OutputSettings": {
    "HlsSettings": {
      "AudioGroupId": "program_audio",
      "AudioRenditionSets": "program_audio",
      "SegmentModifier": "$dt$",
      "IFrameOnlyManifest": "EXCLUDE"
    }
  },
  "NameModifier": "_360"
},
{
  "ContainerSettings": {
    "Container": "M3U8",
    "M3u8Settings": {
      "AudioFramesPerPes": 4,
      "PcrControl": "PCR_EVERY_PES_PACKET",
      "PmtPid": 480,
      "PrivateMetadataPid": 503,
      "ProgramNumber": 1,
      "PatInterval": 0,
      "PmtInterval": 0,
      "TimedMetadata": "NONE",
      "TimedMetadataPid": 502,
      "VideoPid": 481,
      "AudioPids": [
        482,
        483,
        484,
        485,
        486,
        487,
        488,
        489,
        490,
        491,
        492
      ]
    }
  },
  "VideoDescription": {
```

```
"Width": 960,
"ScalingBehavior": "DEFAULT",
"Height": 540,
"TimecodeInsertion": "DISABLED",
"AntiAlias": "ENABLED",
"Sharpness": 50,
"CodecSettings": {
  "Codec": "H_264",
  "H264Settings": {
    "InterlaceMode": "PROGRESSIVE",
    "NumberReferenceFrames": 3,
    "Syntax": "DEFAULT",
    "Softness": 0,
    "GopClosedCadence": 1,
    "GopSize": 2,
    "Slices": 1,
    "GopBReference": "DISABLED",
    "MaxBitrate": 3500000,
    "SlowPal": "DISABLED",
    "SpatialAdaptiveQuantization": "ENABLED",
    "TemporalAdaptiveQuantization": "ENABLED",
    "FlickerAdaptiveQuantization": "DISABLED",
    "EntropyEncoding": "CABAC",
    "FramerateControl": "INITIALIZE_FROM_SOURCE",
    "RateControlMode": "QVBR",
    "CodecProfile": "MAIN",
    "Telecine": "NONE",
    "MinIInterval": 0,
    "AdaptiveQuantization": "HIGH",
    "CodecLevel": "AUTO",
    "FieldEncoding": "PAFF",
    "SceneChangeDetect": "TRANSITION_DETECTION",
    "QualityTuningLevel": "SINGLE_PASS_HQ",
    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
    "UnregisteredSeiTimecode": "DISABLED",
    "GopSizeUnits": "SECONDS",
    "ParControl": "INITIALIZE_FROM_SOURCE",
    "NumberBFramesBetweenReferenceFrames": 2,
    "RepeatPps": "DISABLED"
  }
},
"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
```

```
    "ColorMetadata": "INSERT"
  },
  "OutputSettings": {
    "HlsSettings": {
      "AudioGroupId": "program_audio",
      "AudioRenditionSets": "program_audio",
      "SegmentModifier": "$dt$",
      "IFrameOnlyManifest": "EXCLUDE"
    }
  },
  "NameModifier": "_540"
},
{
  "ContainerSettings": {
    "Container": "M3U8",
    "M3u8Settings": {
      "AudioFramesPerPes": 4,
      "PcrControl": "PCR_EVERY_PES_PACKET",
      "PmtPid": 480,
      "PrivateMetadataPid": 503,
      "ProgramNumber": 1,
      "PatInterval": 0,
      "PmtInterval": 0,
      "TimedMetadata": "NONE",
      "VideoPid": 481,
      "AudioPids": [
        482,
        483,
        484,
        485,
        486,
        487,
        488,
        489,
        490,
        491,
        492
      ]
    }
  }
},
  "VideoDescription": {
    "Width": 1280,
    "ScalingBehavior": "DEFAULT",
    "Height": 720,
```

```
"TimecodeInsertion": "DISABLED",
"AntiAlias": "ENABLED",
"Sharpness": 50,
"CodecSettings": {
  "Codec": "H_264",
  "H264Settings": {
    "InterlaceMode": "PROGRESSIVE",
    "NumberReferenceFrames": 3,
    "Syntax": "DEFAULT",
    "Softness": 0,
    "GopClosedCadence": 1,
    "GopSize": 2,
    "Slices": 1,
    "GopBReference": "DISABLED",
    "MaxBitrate": 5000000,
    "SlowPal": "DISABLED",
    "SpatialAdaptiveQuantization": "ENABLED",
    "TemporalAdaptiveQuantization": "ENABLED",
    "FlickerAdaptiveQuantization": "DISABLED",
    "EntropyEncoding": "CABAC",
    "FramerateControl": "INITIALIZE_FROM_SOURCE",
    "RateControlMode": "QVBR",
    "CodecProfile": "MAIN",
    "Telecine": "NONE",
    "MinIInterval": 0,
    "AdaptiveQuantization": "HIGH",
    "CodecLevel": "AUTO",
    "FieldEncoding": "PAFF",
    "SceneChangeDetect": "TRANSITION_DETECTION",
    "QualityTuningLevel": "SINGLE_PASS_HQ",
    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
    "UnregisteredSeiTimecode": "DISABLED",
    "GopSizeUnits": "SECONDS",
    "ParControl": "INITIALIZE_FROM_SOURCE",
    "NumberBFramesBetweenReferenceFrames": 2,
    "RepeatPps": "DISABLED"
  }
},
"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
"ColorMetadata": "INSERT"
},
"OutputSettings": {
```

```

    "HlsSettings": {
      "AudioGroupId": "program_audio",
      "AudioRenditionSets": "program_audio",
      "SegmentModifier": "$dt$",
      "IFrameOnlyManifest": "EXCLUDE"
    }
  },
  "NameModifier": "_720"
},
{
  "ContainerSettings": {
    "Container": "M3U8",
    "M3u8Settings": {}
  },
  "AudioDescriptions": [
    {
      "AudioSourceName": "Audio Selector 1",
      "CodecSettings": {
        "Codec": "AAC",
        "AacSettings": {
          "Bitrate": 96000,
          "CodingMode": "CODING_MODE_2_0",
          "SampleRate": 48000
        }
      }
    }
  ],
  "OutputSettings": {
    "HlsSettings": {
      "AudioGroupId": "program_audio",
      "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT"
    }
  },
  "NameModifier": "_audio"
}
],
"OutputGroupSettings": {
  "Type": "HLS_GROUP_SETTINGS",
  "HlsGroupSettings": {
    "ManifestDurationFormat": "INTEGER",
    "SegmentLength": 6,
    "TimedMetadataId3Period": 10,
    "CaptionLanguageSetting": "OMIT",
    "Destination": "s3://EXAMPLE-BUCKET/HLS/"
  }
}

```

```
    "DestinationSettings": {
      "S3Settings": {
        "AccessControl": {
          "CannedAcl": "PUBLIC_READ"
        }
      }
    },
    "TimedMetadataId3Frame": "PRIV",
    "CodecSpecification": "RFC_4281",
    "OutputSelection": "MANIFESTS_AND_SEGMENTS",
    "ProgramDateTimePeriod": 600,
    "MinSegmentLength": 0,
    "DirectoryStructure": "SINGLE_DIRECTORY",
    "ProgramDateTime": "EXCLUDE",
    "SegmentControl": "SEGMENTED_FILES",
    "ManifestCompression": "NONE",
    "ClientCache": "ENABLED",
    "StreamInfResolution": "INCLUDE"
  }
},
{
  "CustomName": "MP4",
  "Name": "File Group",
  "Outputs": [
    {
      "ContainerSettings": {
        "Container": "MP4",
        "Mp4Settings": {
          "CslgAtom": "INCLUDE",
          "FreeSpaceBox": "EXCLUDE",
          "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
        }
      },
      "VideoDescription": {
        "Width": 1280,
        "ScalingBehavior": "DEFAULT",
        "Height": 720,
        "TimecodeInsertion": "DISABLED",
        "AntiAlias": "ENABLED",
        "Sharpness": 100,
        "CodecSettings": {
          "Codec": "H_264",
          "H264Settings": {
```



```
    "InterlaceMode": "PROGRESSIVE",
    "ParNumerator": 1,
    "NumberReferenceFrames": 3,
    "Syntax": "DEFAULT",
    "Softness": 0,
    "GopClosedCadence": 1,
    "HrdBufferInitialFillPercentage": 90,
    "GopSize": 2,
    "Slices": 2,
    "GopBReference": "ENABLED",
    "HrdBufferSize": 10000000,
    "MaxBitrate": 5000000,
    "ParDenominator": 1,
    "EntropyEncoding": "CABAC",
    "RateControlMode": "QVBR",
    "CodecProfile": "HIGH",
    "MinIInterval": 0,
    "AdaptiveQuantization": "AUTO",
    "CodecLevel": "AUTO",
    "FieldEncoding": "PAFF",
    "SceneChangeDetect": "ENABLED",
    "QualityTuningLevel": "SINGLE_PASS_HQ",
    "UnregisteredSeiTimecode": "DISABLED",
    "GopSizeUnits": "SECONDS",
    "ParControl": "SPECIFIED",
    "NumberBFramesBetweenReferenceFrames": 3,
    "RepeatPps": "DISABLED",
    "DynamicSubGop": "ADAPTIVE"
  }
},
"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
"ColorMetadata": "INSERT"
},
"AudioDescriptions": [
  {
    "AudioTypeControl": "FOLLOW_INPUT",
    "AudioSourceName": "Audio Selector 1",
    "CodecSettings": {
      "Codec": "AAC",
      "AacSettings": {
        "AudioDescriptionBroadcasterMix": "NORMAL",
        "Bitrate": 160000,

```

```

        "RateControlMode": "CBR",
        "CodecProfile": "LC",
        "CodingMode": "CODING_MODE_2_0",
        "RawFormat": "NONE",
        "SampleRate": 48000,
        "Specification": "MPEG4"
    }
},
"LanguageCodeControl": "FOLLOW_INPUT",
"AudioType": 0
}
]
}
],
"OutputGroupSettings": {
    "Type": "FILE_GROUP_SETTINGS",
    "FileGroupSettings": {
        "Destination": "s3://EXAMPLE-BUCKET/MP4/",
        "DestinationSettings": {
            "S3Settings": {
                "AccessControl": {
                    "CannedAcl": "PUBLIC_READ"
                }
            }
        }
    }
}
},
{
    "CustomName": "Thumbnails",
    "Name": "File Group",
    "Outputs": [
        {
            "ContainerSettings": {
                "Container": "RAW"
            },
            "VideoDescription": {
                "Width": 1280,
                "ScalingBehavior": "DEFAULT",
                "Height": 720,
                "TimecodeInsertion": "DISABLED",
                "AntiAlias": "ENABLED",
                "Sharpness": 50,
                "CodecSettings": {

```

```
        "Codec": "FRAME_CAPTURE",
        "FrameCaptureSettings": {
            "FramerateNumerator": 1,
            "FramerateDenominator": 5,
            "MaxCaptures": 500,
            "Quality": 80
        }
    },
    "AfdSignaling": "NONE",
    "DropFrameTimecode": "ENABLED",
    "RespondToAfd": "NONE",
    "ColorMetadata": "INSERT"
}
],
"OutputGroupSettings": {
    "Type": "FILE_GROUP_SETTINGS",
    "FileGroupSettings": {
        "Destination": "s3://EXAMPLE-BUCKET/Thumbnails/",
        "DestinationSettings": {
            "S3Settings": {
                "AccessControl": {
                    "CannedAcl": "PUBLIC_READ"
                }
            }
        }
    }
}
},
"AdAvailOffset": 0,
"Inputs": [
    {
        "AudioSelectors": {
            "Audio Selector 1": {
                "Offset": 0,
                "DefaultSelection": "DEFAULT",
                "ProgramSelection": 1
            }
        },
        "VideoSelector": {
            "ColorSpace": "FOLLOW"
        },
        "FilterEnable": "AUTO",
```

```
    "PsiControl": "USE_PSI",
    "FilterStrength": 0,
    "DeblockFilter": "DISABLED",
    "DenoiseFilter": "DISABLED",
    "TimecodeSource": "EMBEDDED",
    "FileInput": "s3://EXAMPLE-INPUT-BUCKET/input.mp4"
  }
]
```

- Erstellen Sie im `batch-transcode`-Ordner eine Datei mit einer Lambda-Funktion. Sie können das folgende Python-Beispiel verwenden und die Datei `convert.py` nennen.

S3-Batchvorgänge senden bestimmte Aufgabendaten an eine Lambda-Funktion und erfordern Ergebnisdaten zurück. Anforderungs- und Antwortbeispielen für die Lambda-Funktion, Informationen über Antwort- und Ergebnis-Codes sowie Beispiel-Lambda-Funktionen für S3-Batchvorgänge finden Sie unter [Aufrufen AWS Lambda function](#).

```
import json
import os
from urllib.parse import urlparse
import uuid
import boto3

"""
When you run an S3 Batch Operations job, your job
invokes this Lambda function. Specifically, the Lambda function is
invoked on each video object listed in the manifest that you specify
for the S3 Batch Operations job in Step 5.

Input parameter "event": The S3 Batch Operations event as a request
                        for the Lambda function.

Input parameter "context": Context about the event.

Output: A result structure that Amazon S3 uses to interpret the result
        of the operation. It is a job response returned back to S3 Batch
        Operations.
"""
def handler(event, context):

    invocation_schema_version = event['invocationSchemaVersion']
    invocation_id = event['invocationId']
```

```
task_id = event['tasks'][0]['taskId']

source_s3_key = event['tasks'][0]['s3Key']
source_s3_bucket = event['tasks'][0]['s3BucketArn'].split(':::')[0]
source_s3 = 's3://' + source_s3_bucket + '/' + source_s3_key

result_list = []
result_code = 'Succeeded'
result_string = 'The input video object was converted successfully.'

# The type of output group determines which media players can play
# the files transcoded by MediaConvert.
# For more information, see Creating outputs with AWS Elemental MediaConvert.
output_group_type_dict = {
    'HLS_GROUP_SETTINGS': 'HlsGroupSettings',
    'FILE_GROUP_SETTINGS': 'FileGroupSettings',
    'CMAF_GROUP_SETTINGS': 'CmafGroupSettings',
    'DASH_ISO_GROUP_SETTINGS': 'DashIsoGroupSettings',
    'MS_SMOOTH_GROUP_SETTINGS': 'MsSmoothGroupSettings'
}

try:
    job_name = 'Default'
    with open('job.json') as file:
        job_settings = json.load(file)

    job_settings['Inputs'][0]['FileInput'] = source_s3

    # The path of each output video is constructed based on the values of
    # the attributes in each object of OutputGroups in the job.json file.
    destination_s3 = 's3://{0}/{1}/{2}' \
        .format(os.environ['DestinationBucket'],
                os.path.splitext(os.path.basename(source_s3_key))[0],
                os.path.splitext(os.path.basename(job_name))[0])

    for output_group in job_settings['OutputGroups']:
        output_group_type = output_group['OutputGroupSettings']['Type']
        if output_group_type in output_group_type_dict.keys():
            output_group_type = output_group_type_dict[output_group_type]
            output_group['OutputGroupSettings'][output_group_type]
['Destination'] = \
                "{0}{1}".format(destination_s3,
                                urlparse(output_group['OutputGroupSettings']
[output_group_type]['Destination']).path)
```

```
        else:
            raise ValueError("Exception: Unknown Output Group Type {}".format(output_group_type))

    job_metadata_dict = {
        'assetID': str(uuid.uuid4()),
        'application': os.environ['Application'],
        'input': source_s3,
        'settings': job_name
    }

    region = os.environ['AWS_DEFAULT_REGION']
    endpoints = boto3.client('mediaconvert', region_name=region) \
        .describe_endpoints()
    client = boto3.client('mediaconvert', region_name=region,
                          endpoint_url=endpoints['Endpoints'][0]['Url'],
                          verify=False)

    try:
        client.create_job(Role=os.environ['MediaConvertRole'],
                          UserMetadata=job_metadata_dict,
                          Settings=job_settings)
        # You can customize error handling based on different error codes that
        # MediaConvert can return.
        # For more information, see MediaConvert error codes.
        # When the result_code is TemporaryFailure, S3 Batch Operations retries
        # the task before the job is completed. If this is the final retry,
        # the error message is included in the final report.
    except Exception as error:
        result_code = 'TemporaryFailure'
        raise

except Exception as error:
    if result_code != 'TemporaryFailure':
        result_code = 'PermanentFailure'
    result_string = str(error)

finally:
    result_list.append({
        'taskId': task_id,
        'resultCode': result_code,
        'resultString': result_string,
    })
```

```
return {
    'invocationSchemaVersion': invocation_schema_version,
    'treatMissingKeyAs': 'PermanentFailure',
    'invocationId': invocation_id,
    'results': result_list
}
```

- Um ein Bereitstellungspaket mit `convert.py` und `job.json` als `.zip`-Datei mit dem Namen `lambda.zip` zu erstellen, öffnen Sie in Ihrem lokalen Terminal den `batch-transcode`-Ordner, den Sie bereits erstellt haben, und führen Sie den folgenden Befehl aus.

Führen Sie für macOS-Benutzer den folgenden Befehl aus:

```
zip -r lambda.zip convert.py job.json
```

Führen Sie für Windows-Benutzer die folgenden Befehle aus:

```
powershell Compress-Archive convert.py lambda.zip
```

```
powershell Compress-Archive -update job.json lambda.zip
```

Erstellen einer Lambda-Funktion mit einer Ausführungsrolle (Konsole)

- Öffnen Sie AWS Lambda Konsole bei <https://console.aws.amazon.com/lambda/>.
- Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
- Wählen Sie Create function (Funktion erstellen).
- Wählen Sie Author from scratch aus.
- Führen Sie unter Basic information (Grundlegende Informationen) die folgenden Schritte aus:
 - Geben Sie für Function name (Funktionsname) **tutorial-lambda-convert** ein.
 - Wählen Sie für Laufzeit Python 3.8 oder eine höhere Version von Python aus.
- Wählen Sie Ändern der standardmäßigen Ausführungsrolle und wählen Sie unter Ausführungsrolle Verwenden einer vorhandenen Rolle.
- Wählen Sie unter Existierende Rolle den Namen der IAM Rolle aus, die Sie in [Schritt 3](#) für Ihre Lambda-Funktion erstellt haben (z. B. **tutorial-lambda-transcode-role**).

8. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.
9. Wählen Sie Funktion erstellen aus.

Stellen Sie Ihre Lambda-Funktion mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion (Konsole)

1. Wählen Sie im Abschnitt Code-Quelle der Seite für die Lambda-Funktion, die Sie erstellt haben (z. B. **tutorial-lambda-convert**), Hochladen von und dann ZIP-Datei aus.
2. Wählen Sie Upload (Hochladen) aus, um Ihre lokale `.zip`-Datei auszuwählen.
3. Wählen Sie die zuvor erstellte `lambda.zip`-Datei und klicken Sie auf Öffnen.
4. Wählen Sie Speichern aus.
5. Wählen Sie im Abschnitt Runtime settings (Laufzeiteinstellungen) die Option Bearbeiten aus.
6. Um der Lambda-Laufzeit mitzuteilen, welche Handler-Methode in Ihrem Lambda-Funktionscode aufgerufen werden soll, geben Sie **convert.handler** in das Feld Handler ein.

Wenn Sie eine Funktion in Python konfigurieren, besteht der Wert der Handler-Einstellung aus dem Dateinamen und dem Namen des Handler-Moduls, getrennt durch einen Punkt (`.`). Beispielsweise ruft `convert.handler` die Methode `handler` auf, die in der Datei `convert.py` definiert ist.

7. Wählen Sie Save (Speichern) aus.
8. Wählen Sie auf der Seite der Lambda-Funktion den Tab Konfiguration. Im linken Navigationsbereich auf der Registerkarte Konfiguration wählen Sie Umgebungsvariablen und wählen dann Bearbeiten aus.
9. Wählen Sie Umgebungsvariablen hinzufügen aus. Geben Sie dann den angegebenen Schlüssel und Wert für jede der folgenden Umgebungsvariablen ein:

- Schlüssel: **DestinationBucket** Wert: **tutorial-bucket-2**

Dieser Wert ist der S3 Bucket für Ausgabe-Mediendateien, die Sie in [Schritt 1](#) erstellt haben.

- Schlüssel: **MediaConvertRole** Wert: **arn:aws:iam::111122223333:role/tutorial-mediaconvert-role**

Dieser Wert entspricht ARN der IAM Rolle `MediaConvert`, für die Sie in [Schritt 2](#) erstellt haben. Stellen Sie sicher, dass Sie diesen ARN Wert durch den tatsächlichen ARN Wert Ihrer IAM Rolle ersetzen.

- Schlüssel: **Application** Wert: **Batch-Transcoding**

Dieser Wert ist der Name der Anwendung.

10. Wählen Sie Save (Speichern) aus.
11. (Optional) Wählen Sie im Tab Konfiguration im Abschnitt Allgemeine Konfiguration des linken Navigationsbereichs Bearbeiten aus. Geben Sie im Timeout-Feld **2 m 0 s** ein. Wählen Sie dann Save (Speichern) aus.

Timeout ist die Zeitspanne, die Lambda einer Funktion für einen Aufruf zulässt, bevor diese gestoppt wird. Der Standardwert ist 3 Sekunden. Die Preise basieren auf dem konfigurierten Arbeitsspeicher und der Zeit, für die der Code ausgeführt wird. Weitere Informationen finden Sie unter [AWS Lambda Preisgestaltung](#).

Schritt 5: Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket

Nachdem Sie die Transcodierungs-Lambda-Funktion eingerichtet haben, erstellen Sie einen S3-Batchvorgangsauftrag, um eine Reihe von Videos zu transcodieren. Zuerst benötigen Sie eine Liste von Eingabe-Videoobjekten, für die S3-Batch-Vorgänge die angegebene Transcodierungsaktion ausführen soll. Um eine Liste von Eingabe-Videoobjekten abzurufen, können Sie einen S3-Inventarbericht für Ihren S3-Quell-Bucket erstellen (z. B. **tutorial-bucket-1**).

Teilschritte

- [Erstellen und Konfigurieren eines Buckets für S3-Inventarberichte für Eingabevideos](#)
- [Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket](#)
- [Überprüfen Sie den Lagerbestandsbericht für Ihren S3-Videoquellen-Bucket](#)

Erstellen und Konfigurieren eines Buckets für S3-Inventarberichte für Eingabevideos

Um einen S3-Bestandslistenbericht zu speichern, der die Objekte des S3-Quell-Buckets auflistet, erstellen Sie einen S3-Inventarziel-Bucket und konfigurieren Sie dann eine Bucket-Richtlinie, damit der Bucket Inventardateien in den S3-Quell-Bucket schreibt.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

4. Geben Sie im Feld Bucket Name einen Namen für Ihren Bucket ein (z. B. **tutorial-bucket-3**).

5. Für AWS-Region, wählen Sie AWS-Region wo sich der Eimer befinden soll.

Der Ziel-Bucket für das Inventar muss sich im selben Bucket befinden AWS-Region wie der Quell-Bucket, in dem Sie S3 Inventory einrichten. Der Ziel-Bucket für das Inventar kann sich in einem anderen befinden AWS-Konto.

6. Behalten Sie die Einstellungen für öffentlichen Zugriff für diesen Bucket blockieren auf die Standardeinstellungen gestellt (Alle öffentlichen Zugriffe blockieren ist aktiviert).

7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

8. Wählen Sie Bucket erstellen aus.

9. Wählen Sie in der Liste der Buckets den Namen des Buckets, den Sie gerade erstellt haben (z. B. **tutorial-bucket-3**).

10. Um Amazon S3 die Berechtigung zu erteilen, Daten für die Bestandsberichte in den S3-Basisziel-Bucket zu schreiben, wählen Sie das Tab Berechtigungen aus.

11. Scrollen Sie nach unten zum Abschnitt Bucket-Richtlinie und wählen Sie Bearbeiten aus. Die Seite Bucket-Richtlinie wird geöffnet.

12. Um Berechtigungen für S3-Inventar zu erteilen, fügen Sie im Feld Richtlinie die folgende Bucket-Richtlinie ein.

Ersetzen Sie die drei Beispielwerte durch die folgenden Werte:

- Der Name des Buckets, den Sie erstellt haben, um die Bestandsberichte zu speichern (z. B. *tutorial-bucket-3*).
- Der Name des Quell-Buckets, in dem die Eingabe-Videos gespeichert werden (z. B. *tutorial-bucket-1*).
- Das Tool AWS-Konto ID, mit der Sie den S3-Videoquellen-Bucket erstellt haben (z. *111122223333* B.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "InventoryAndAnalyticsExamplePolicy",
      "Effect": "Allow",
      "Principal": {"Service": "s3.amazonaws.com"},
```

```
"Action": "s3:PutObject",
"Resource": ["arn:aws:s3:::tutorial-bucket-3/*"],
"Condition": {
  "ArnLike": {
    "aws:SourceArn": "arn:aws:s3:::tutorial-bucket-1"
  },
  "StringEquals": {
    "aws:SourceAccount": "111122223333",
    "s3:x-amz-acl": "bucket-owner-full-control"
  }
}
]
```

13. Wählen Sie Änderungen speichern.

Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket

Sie müssen S3-Inventar für Ihren S3-Videoquellen-Bucket konfigurieren, um eine flache Dateiliste mit Videoobjekten und Metadaten zu generieren. Diese geplanten Bestandsberichte können alle Objekte im Bucket oder Objekte enthalten, die nach einem gemeinsamen Präfix gruppiert sind. In diesem Tutorial enthält der S3-Bestandsbericht alle Video-Objekte in Ihrem S3-Quell-Bucket.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Um einen S3-Inventarbericht der Eingabe-Videos in Ihrem S3-Quell-Bucket zu konfigurieren, wählen Sie in der Liste Buckets zum Beispiel den Namen des S3-Quell-Buckets aus (z. B. **tutorial-bucket-1**).
4. Wählen Sie den Tab Management.
5. Blättern Sie nach unten bis zur Konfigurationen für Bestandserfassung und wählen Sie Erstellen der Bestandskonfiguration aus.
6. Geben Sie für Name der Bestandserfassung z. B. einen Namen ein (z. B. **tutorial-inventory-config**).
7. Wählen Sie unter Bestandserfassung Nur aktuelle Version für Objekt-Versionen und lassen Sie andere Einstellungen der Bestandserfassung auf die Standardwerte für dieses Tutorial eingestellt.

8. Wählen Sie im Abschnitt Berichtsdetails für Ziel-Bucket Dieses Konto aus.
9. Wählen Sie für Ziel S3 durchsuchen und wählen Sie den Ziel-Bucket aus, den Sie bereits erstellt haben, um die Lagerbestandsberichte zu speichern (z. B. **tutorial-bucket-3**). Wählen Sie dann Pfad wählen aus.

Der Ziel-Bucket für das Inventar muss sich im selben Bucket befinden AWS-Region wie der Quell-Bucket, in dem Sie S3 Inventory einrichten. Der Ziel-Bucket für das Inventar kann sich in einem anderen befinden AWS-Konto.

Unter dem Bucket-Feld Destination (Ziel) wird die Destination bucket permission (Ziel-Bucket-Berechtigung) der Bestands-Ziel-Bucket-Richtlinie hinzugefügt, damit Amazon S3 Daten in diesen Bestands-Ziel-Bucket platzieren kann. Weitere Informationen finden Sie unter [Erstellen einer Ziel-Bucket-Richtlinie](#).

10. Wählen Sie für Häufigkeit Täglich aus.
11. Wählen Sie als Ausgabeformat die Option CSV.
12. Wählen Sie für Status die Option Aktiviert.
13. Wählen Sie im Abschnitt Serverseitige Verschlüsselung Deaktivieren für dieses Tutorial.

Weitere Informationen erhalten Sie unter [Konfigurieren des Bestands mit der S3-Konsole](#) und [Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung](#).

14. Wählen Sie unter Zusätzliche Felder – optional Größe, Letzte Änderung, und Speicherklasse aus.
15. Wählen Sie Create (Erstellen) aus.


Weitere Informationen finden Sie unter [Konfigurieren des Bestands mit der S3-Konsole](#).

Überprüfen Sie den Lagerbestandsbericht für Ihren S3-Videoquellen-Bucket

Wenn ein Bestandsbericht veröffentlicht wird, werden die Manifestdateien an den S3-Bestands-Ziel-Bucket gesendet.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.

3. Wählen Sie in der Liste Buckets den Namen des Video-Quell-Buckets (z. B. **tutorial-bucket-1**).
4. Wählen Sie Management (Verwaltung) aus.
5. Um zu sehen, ob Ihr S3-Bestandsbericht für das Erstellen eines S3-Batchvorgänge-Auftrags in [Schritt 7](#) bereit ist, überprüfen Sie unter Bestandskonfigurationen, ob die Schaltfläche Auftrag aus Manifest erstellen aktiviert ist.

 Note

Es kann bis zu 48 Stunden dauern, bis der erste Bestandsbericht bereitgestellt wird. Wenn die Schaltfläche Auftrag aus Manifest erstellendeaktiviert ist, wurde der erste Bestandsbericht nicht zugestellt. Warten Sie, bis der erste Bestandsbericht zugestellt wurde und die Schaltfläche Auftrag aus Manifest erstellen aktiviert ist, bevor Sie einen S3-Batchvorgänge-Auftrag in [Schritt 7](#) erstellen.

6. Um einen S3-Bestandsbericht (`manifest.json`) in der die Spalte Ziel zu überprüfen, wählen Sie den Namen des Bestandsziels aus, den Sie bereits zum Speichern von Bestandsberichten erstellt haben (z. B. **tutorial-bucket-3**).
7. Wählen Sie auf der Registerkarte Objekte den vorhandenen Ordner mit dem Namen Ihres S3-Quell-Buckets (z. B. **tutorial-bucket-1**). Wählen Sie dann den Namen aus, den Sie in Name der Bestandserfassung eingegeben haben, als Sie die Inventarkonfiguration früher erstellt haben (z. B. **tutorial-inventory-config**).

Sie können eine Liste von Ordnern mit den Generierungsdaten der Berichte als Namen sehen.

8. Um den täglichen S3-Bestandsbericht an einem bestimmten Datum zu überprüfen, wählen Sie den Ordner mit dem entsprechenden Namen des Generierungsdatums aus und wählen Sie dann `manifest.json` aus.
9. Um die Details des Bestandsberichts zu einem bestimmten Datum zu überprüfen, wählen Sie auf der Seite `manifest.json` Herunterladen oder Öffnen aus.

Schritt 6: Eine IAM Rolle für S3 Batch Operations erstellen

Um S3 Batch Operations für die Batch-Transcodierung zu verwenden, müssen Sie zunächst eine IAM Rolle erstellen, um Amazon S3 Berechtigungen zur Durchführung von S3-Batch-Vorgängen zu erteilen.

Teilschritte

- [Erstellen Sie eine IAM Richtlinie für S3 Batch Operations](#)
- [Erstellen Sie eine IAM S3-Rolle für Batch Operations und fügen Sie Berechtigungsrichtlinien hinzu](#)

Erstellen Sie eine IAM Richtlinie für S3 Batch Operations

Sie müssen eine IAM Richtlinie erstellen, die S3 Batch Operations die Erlaubnis erteilt, das Eingabemanifest zu lesen, die Lambda-Funktion aufzurufen und den Bericht zum Abschluss des S3 Batch Operations-Jobs zu schreiben.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich die Option Policies (Richtlinien) aus.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie die JSONRegisterkarte.
5. Fügen Sie die folgende JSON Richtlinie in das JSOTextfeld ein.

Ersetzen Sie in der JSON Richtlinie die vier Beispielwerte durch die folgenden Werte:

- Der Name des Quell-Buckets, in dem Ihre Eingabe-Videos gespeichert werden (z. B. *tutorial-bucket-1*).
- Der Name des Bestands-Ziel-Buckets, den Sie in [Schritt 5](#) erstellt haben, um manifest.json-Dateien (z. B. *tutorial-bucket-3*) zu speichern.
- Der Name des Buckets, den Sie in [Schritt 1](#) erstellt haben, um Ausgabe-Mediendateien zu speichern (z. B. *tutorial-bucket-2*). In diesem Tutorial legen wir Auftragsvervollständigungsberichte in den Ziel-Bucket für Ausgabe-Mediendateien.
- Die Rolle ARN der Lambda-Funktion, die Sie in [Schritt 4](#) erstellt haben. Gehen Sie wie folgt vor, um die Rolle ARN der Lambda-Funktion zu finden und zu kopieren:
 - Öffnen Sie in einem neuen Browser-Tab die Seite Funktionen der Lambda-Konsole bei <https://console.aws.amazon.com/lambda/home#/functions>.
 - Wählen Sie in der Liste Funktionen den Namen der Lambda-Funktion aus, die Sie in [Schritt 4](#) (zum Beispiel **tutorial-lambda-convert**) erstellt haben.
 - Wählen Sie die Option CopyARN Kopieren aus.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3Get",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::tutorial-bucket-1/*",
        "arn:aws:s3:::tutorial-bucket-3/*"
      ]
    },
    {
      "Sid": "S3PutJobCompletionReport",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::tutorial-bucket-2/*"
    },
    {
      "Sid": "S3BatchOperationsInvokeLambda",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:us-west-2:111122223333:function:tutorial-lambda-convert"
      ]
    }
  ]
}
```

6. Wählen Sie Next: Tags (Weiter: Tags) aus.
7. Klicken Sie auf Weiter: Prüfen.
8. Geben Sie im Feld Name **tutorial-s3batch-policy** ein.
9. Wählen Sie Create Policy (Richtlinie erstellen) aus.

Erstellen Sie eine IAM S3-Rolle für Batch Operations und fügen Sie Berechtigungsrichtlinien hinzu

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen) und dann Create Role (Rolle erstellen) aus.
3. Wählen Sie das Symbol AWS-ServiceRollentyp und wählen Sie dann den S3-Dienst aus.
4. Wählen Sie unter Ihren Anwendungsfall auswählen S3-Batch-Vorgänge.
5. Wählen Sie Weiter: Berechtigungen aus.
6. Geben Sie unter Zugriffsrichtlinien anhängen den Namen der IAM Richtlinie, die Sie zuvor erstellt haben (z. B. **tutorial-s3batch-policy**), in das Suchfeld ein, um die Liste der Richtlinien zu filtern. Aktivieren Sie das Kontrollkästchen neben dem Namen der Richtlinie (z. B. **tutorial-s3batch-policy**).
7. Wählen Sie Next: Tags (Weiter: Tags) aus.
8. Wählen Sie Weiter: Prüfen aus.
9. Geben Sie für Role name (Rollenname) den Namen **tutorial-s3batch-role** ein.
10. Wählen Sie Create role (Rolle erstellen) aus.

Nachdem Sie die IAM Rolle für S3 Batch Operations erstellt haben, wird die folgende Vertrauensrichtlinie automatisch an die Rolle angehängt. Diese Vertrauensrichtlinie ermöglicht es dem Dienstprinzipal von S3 Batch Operations, die IAM Rolle zu übernehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batchoperations.s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```


Schritt 7: Einrichten und Ausführen eines Auftrags für S3-Batchvorgänge

Um einen S3-Batchvorgänge-Auftrag zu erstellen, um die Eingabevideos in Ihrem S3-Quell-Bucket zu verarbeiten, müssen Sie Parameter für diesen bestimmten Auftrag angeben.

Note

Um mit der Erstellung eines S3-Batchvorgänge-Auftrags zu beginnen, stellen Sie sicher, dass die Schaltfläche Auftrag aus Manifest erstellen aktiviert ist. Weitere Informationen finden Sie unter [Überprüfen Sie den Lagerbestandsbericht für Ihren S3-Videoquellen-Bucket](#). Wenn die Schaltfläche Auftrag aus Manifest erstellen deaktiviert ist, wurde der erste Inventarbericht nicht übermittelt und Sie müssen warten, bis die Schaltfläche aktiviert ist. Nachdem Sie Amazon S3-Bestand für Ihren S3-Quell-Bucket in [Schritt 5](#) konfiguriert haben, kann es bis zu 48 Stunden dauern, bis der erste Bestandsbericht übermittelt wird.

Teilschritte

- [Erstellen eines S3-Batchvorgangsauftrags](#)
- [Führen Sie den S3-Batchvorgänge-Auftrag aus, um Ihre Lambda-Funktion aufzurufen.](#)
- [\(Optional\) Prüfen Sie Ihren Abschlussbericht](#)
- [\(Optional\) Überwachen Sie jeden Lambda-Aufruf in der Lambda-Konsole](#)
- [\(Optional\) Überwachen Sie jeden MediaConvert Video-Transcodierungsauftrag in der Konsole MediaConvert](#)

Erstellen eines S3-Batchvorgangsauftrags

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie Job erstellen aus.
4. Für AWS-Region, wählen Sie die Region aus, in der Sie Ihren Job erstellen möchten.

In diesem Tutorial müssen Sie den S3-Batchvorgänge-Auftrag zum Aufrufen einer Lambda-Funktion in derselben Region wie den S3-Videoquellen-Bucket erstellen, in dem sich die im Manifest referenzierten Objekte befinden.

5. Gehen Sie im Abschnitt Manifest wie folgt vor:
 - a. Wählen Sie für Manifestformat S3-Inventory-Bericht (manifest.json) aus.
 - b. Wählen Sie für Manifest-Objekt S3 durchsuchen, um den Bucket zu finden, den Sie in [Schritt 5](#) zum Speichern von Bestandsberichten (z. B. **tutorial-bucket-3**) erstellt haben. Navigieren Sie auf der Seite Manifest-Objekt durch die Objektnamen, bis Sie eine manifest.json-Datei für ein bestimmtes Datum finden. Diese Datei listet die Informationen zu allen Videos auf, die Sie Batch-Transcodieren möchten. Wenn Sie die manifest.json-Datei gefunden haben, die Sie verwenden möchten, wählen Sie die Optionsschaltfläche daneben. Wählen Sie dann Pfad wählen aus.
 - c. (Optional) Geben Sie für Versions-ID des Manifestobjekts – optional die Versions-ID des Manifestobjekts ein, wenn Sie eine andere Version als die aktuelle verwenden möchten.
6. Wählen Sie Weiter.
7. Um die Lambda-Funktion zu verwenden, um alle in der ausgewählten manifest.json Datei aufgelisteten Objekte zu transkodieren, wählen Sie unter Operationstyp die Option Invoke AWS Lambda Funktion.
8. Führen Sie im Abschnitt Aufrufen einer Lambda-Funktion Folgendes aus:
 - a. Klicken Sie auf Wählen Sie aus den Funktionen Ihres Kontos aus.
 - b. Wählen Sie für Lambda-Funktion die Lambda-Funktion aus, die Sie in [Schritt 4](#) (zum Beispiel **tutorial-lambda-convert**) erstellt haben.
 - c. Behalten Sie für die Lambda-Funktionsversion den Standardwert \$ LATEST bei.
9. Wählen Sie Weiter. Die Seite Konfigurieren zusätzlicher Optionen wird geöffnet.
10. Behalten Sie im Abschnitt Zusätzliche Optionen die Standardeinstellungen bei.

Weitere Informationen zu diesen Optionen finden Sie unter [Batch-Vorgangsauftrag-Anforderungselemente](#).

11. Wählen Sie im Abschnitt Abschlussbericht für Pfad zum Ziel des Abschlussberichts S3 durchsuchen aus. Finden Sie den Bucket, den Sie in [Schritt 1](#) für die Ausgabe-Mediendateien erstellt haben (z. B. **tutorial-bucket-2**). Wählen Sie das Optionsfeld neben dem Namen dieses Buckets aus. Wählen Sie dann Pfad wählen aus.

Lassen Sie die restlichen Einstellungen des Abschlussberichts auf die Standardwerte eingestellt. Weitere Informationen zu den Einstellungen des Fertigstellungsberichts finden Sie unter [Batch-Vorgangsauftrag-Anforderungselemente](#). Ein Abschlussbericht führt eine Aufzeichnung der Details des Auftrags und der ausgeführten Vorgänge.

12. Wählen Sie im Abschnitt Berechtigungen die Option Aus vorhandenen IAM Rollen auswählen aus. Wählen Sie unter IAMRolle die IAM Rolle für Ihren S3 Batch Operations-Job aus, den Sie in [Schritt 6](#) erstellt haben (z. B. **tutorial-s3batch-role**).
13. Wählen Sie Weiter.
14. Überprüfen Sie die Einstellungen auf der Seite Review. Wählen Sie dann Auftrag erstellen aus.

Nachdem S3 das Manifest Ihres S3-Batchvorgänge-Auftrags gelesen hat, wechselt es den Auftrag in den Zustand *Awaiting your confirmation to run* (Wartet auf Ihre Bestätigung zur Ausführung). Um Aktualisierungen des Auftragsstatus anzuzeigen, aktualisieren Sie die Seite. Sie können Ihren Job erst ausführen, wenn der Status lautet *Wartet auf Ihre Bestätigung zur Ausführung*.

Führen Sie den S3-Batchvorgänge-Auftrag aus, um Ihre Lambda-Funktion aufzurufen.

Führen Sie Ihren Batchvorgänge-Auftrag aus, um Ihre Lambda-Funktion für die Videotranscodierung aufzurufen. Wenn Ihr Auftrag fehlschlägt, können Sie den Abschlussbericht überprüfen, um die Ursache zu ermitteln.

So führen Sie den S3-Batchvorgänge-Auftrag aus

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie aus der Liste Aufträge die Auftrags-ID des Auftrags in der ersten Zeile aus, bei der es sich um den vorher erstellten S3-Batchvorgänge-Auftrag handelt.
4. Wählen Sie Auftrag ausführen aus.
5. Überprüfen Sie die Auftrags-Parameter erneut, und bestätigen Sie, dass der Wert für *Insgesamt im Manifest aufgelistete Objekte* mit der Anzahl der Objekte im Manifest identisch ist. Wählen Sie dann Auftrag ausführen.

Ihre Seite des S3-Batchvorgänge-Auftrags wird geöffnet.

6. Nachdem der Auftrag gestartet wurde, kontrollieren Sie unter Status den Fortschritt Ihres S3-Batchvorgänge-Auftrags, wie z. B. Status, % abgeschlossen, Gesamt erfolgreich (Rate), Gesamt fehlgeschlagen (Rate), Beendigungsdatum, und Grund für die Beendigung.

Wenn der S3-Batchvorgänge-Auftrag abgeschlossen ist, zeigen Sie die Daten auf Ihrer Auftragseite an, um zu bestätigen, dass der Auftrag wie erwartet abgeschlossen wurde.

Wenn mehr als 50 Prozent der Objektvorgänge eines S3-Batchvorgänge-Auftrags fehlschlagen, nachdem mehr als 1000 Vorgänge versucht wurden, schlägt der Auftrag automatisch fehl. Um Ihren Abschlussbericht zu überprüfen, um die Ursache der Fehler zu ermitteln, verwenden Sie das folgenden optionale Verfahren.

(Optional) Prüfen Sie Ihren Abschlussbericht

Sie können Ihren Abschlussbericht verwenden, um festzustellen, welche Objekte fehlgeschlagen sind und welche Ursache der Fehler sind.

Um Ihren Abschlussbericht auf Details zu fehlgeschlagenen Objekten zu überprüfen:

1. Blättern Sie auf der Seite Ihres S3-Batchvorgänge-Auftrags zum Abschnitt Abschlussbericht und wählen Sie den Link unter Ziel des Abschlussberichts aus.

Die Seite des S3-Ausgabe-Ziel-Buckets wird geöffnet.


2. Wählen Sie auf der Registerkarte Objekte auf den Ordner, dessen Name mit der Auftrags-ID des vorher erstellten S3-Batchvorgänge-Auftrags endet.
3. Klicken Sie auf Ergebnis/se.
4. Aktivieren Sie das Kontrollkästchen neben der `.csv`-Datei.
5. Wählen Sie Öffnen oder Herunterladen, um den Auftragsbericht anzuzeigen.

(Optional) Überwachen Sie jeden Lambda-Aufruf in der Lambda-Konsole

Nachdem der S3-Batchvorgänge-Auftrag gestartet wurde, ruft der Auftrag, die Lambda-Funktion für jedes Eingabe-Video-Objekt auf. S3 schreibt Protokolle jedes Lambda-Aufrufs in Logs. CloudWatch Sie können das Monitoring-Dashboard der Lambda-Konsole verwenden, um Ihre Lambda-Funktion zu überwachen.

1. Öffnen Sie AWS Lambda Konsole bei <https://console.aws.amazon.com/lambda/>
2. Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
3. Wählen Sie in der Liste Funktionen den Namen der Lambda-Funktion aus, die Sie in [Schritt 4](#) (zum Beispiel **tutorial-lambda-convert**) erstellt haben.
4. Wählen Sie den Tab Überwachung.
5. Finden Sie unter Metriken die Laufzeitmetriken für Ihre Lambda-Funktion.

6. Sehen Sie sich unter Logs die Protokolldaten für jeden Lambda-Aufruf über CloudWatch Logs Insights an.

 Note

Wenn Sie S3-Batch-Vorgänge mit einer Lambda-Funktion verwenden, wird die Lambda-Funktion für jedes Objekt aufgerufen. Wenn Ihr S3-Batchvorgänge-Auftrag groß ist, kann er mehrere Lambda-Funktionen gleichzeitig aufrufen, was zu einer Spitze der Lambda-Gleichzeitigkeit führt.

Jeder AWS-Konto hat eine Lambda-Parallelitätsquote pro Region. Weitere Informationen finden Sie unter [AWS Lambda Funktionsskalierung](#) in der AWS Lambda Leitfaden für Entwickler. Eine bewährte Methode für die Verwendung von Lambda-Funktionen mit S3-Batchvorgängen besteht darin, eine Gleichzeitigkeitsgrenze für die Lambda-Funktion selbst festzulegen. Ein Parallelitäts-Limit aufzustellen verhindert, dass Ihr Auftrag den größten Teil Ihrer Lambda-Parallelität verbraucht und möglicherweise andere Funktionen in Ihrem Konto drosselt. Weitere Informationen finden Sie unter [Managing Lambda Reserved Concurrency](#) in der AWS Lambda Entwicklerhandbuch.

(Optional) Überwachen Sie jeden MediaConvert Video-Transcodierungsauftrag in der Konsole MediaConvert

Ein MediaConvert Job erledigt die Transcodierung einer Mediendatei. Wenn Ihr S3 Batch Operations-Job Ihre Lambda-Funktion für jedes Video aufruft, erstellt jeder Lambda-Funktionsaufruf einen MediaConvert Transcodierungsauftrag für jedes Eingabevideo.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die MediaConvert Konsole unter <https://console.aws.amazon.com/mediaconvert/>.
2. Wenn die MediaConvert Einführungsseite angezeigt wird, wählen Sie Erste Schritte.
3. Zeige Sie aus der Liste Aufträge jede Zeile an, um die Transcodierungsaufgabe für jedes Eingabe-Video zu überwachen.
4. Geben Sie die Zeile eines Auftrags an, den Sie überprüfen möchten, und wählen Sie den Auftrags-ID-Link, um die Seite Auftrags-Details zu öffnen.
5. Wählen Sie auf der Seite mit der Jobübersicht unter Ausgaben den Link für dieHLS,MP4, oder Miniaturansichten aus, je nachdem, was von Ihrem Browser unterstützt wird, um zum S3-Ziel-Bucket für die Ausgabemediendateien zu gelangen.

6. Wählen Sie im entsprechenden Ordner (HLSMP4, oder Miniaturansichten) Ihres Ziel-Buckets für die S3-Ausgabe den Namen des Objekts der Ausgabemediendatei aus.

Die Detailseite des Objekts wird geöffnet.

7. Wählen Sie auf der Detailseite des Objekts unter Objektübersicht den Link unter Objekt aus, URL um die transkodierte Ausgabemediendatei anzusehen.

Schritt 8: Überprüfen Sie die Ausgabe-Mediendateien aus Ihrem S3-Ziel-Bucket

So überprüfen Sie die Ausgabe-Mediendateien aus Ihrem S3-Ziel-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie aus der Liste Buckets den Namen des S3-Ziel-Buckets für Ausgabe-Mediendateien aus, die Sie im Abschnitt [Schritt 1](#) (zum Beispiel **tutorial-bucket-2**) erstellt haben.
4. Im Tab Objekte enthält jedes Eingabe-Video einen Ordner mit dem Namen des Eingabe-Videos. Jeder Ordner enthält die transcodierten Ausgabe-Mediendateien für ein Eingabe-Video.

Gehen Sie wie folgt vor, um die Ausgabe-Mediendateien für ein Eingabe-Video zu überprüfen:

- a. Wählen Sie den Ordner mit dem Namen des Eingabe-Videos, das Sie überprüfen möchten.
- b. Wählen Sie den Ordner Standard/ aus.
- c. Wählen Sie den Ordner für ein transkodiertes Format (HLS,MP4, oder Miniaturansichten in diesem Tutorial).
- d. Wählen Sie den Namen der Ausgabemediendatei aus.
- e. Um die transkodierte Datei anzusehen, klicken Sie auf der Detailseite des Objekts auf den Link unter Objekt. URL

Ausgabemediendateien in diesem HLS Format sind in kurze Segmente aufgeteilt. Um diese Videos abzuspielen, betten Sie das Objekt URL der .m3u8 Datei in einen kompatiblen Player ein.

Schritt 9: Bereinigen

Wenn Sie Videos mit S3 Batch Operations, Lambda und MediaConvert nur als Lernübung transkodiert haben, löschen Sie die AWS Ressourcen, die Sie so zugewiesen haben, dass Ihnen keine Gebühren mehr anfallen.

Teilschritte

- [Löschen der S3-Bestandskonfiguration für Ihren S3-Quell-Bucket](#)
- [Löschen Sie die Lambda-Funktion](#)
- [Löschen Sie die CloudWatch Protokollgruppe](#)
- [Löschen Sie die IAM Rollen zusammen mit den Inline-Richtlinien für die IAM Rollen](#)
- [Löschen Sie die vom Kunden verwaltete Richtlinie IAM](#)
- [Leeren Sie die S3-Buckets](#)
- [Löschen der S3-Buckets](#)

Löschen der S3-Bestandskonfiguration für Ihren S3-Quell-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen Ihres Quell-Buckets (z. B. **tutorial-bucket-1**).
4. Wählen Sie den Tab Management.
5. Wählen Sie im Abschnitt Bestandskonfigurationen die Optionsschaltfläche neben der Bestandskonfiguration aus, die Sie in [Schritt 5](#) (zum Beispiel **tutorial-inventory-config**) erstellt haben.
6. Wählen Sie Löschen und dann Bestätigen aus.

Löschen Sie die Lambda-Funktion

1. Öffnen Sie AWS Lambda Konsole unter <https://console.aws.amazon.com/lambda/>.
2. Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
3. Aktivieren Sie das Kontrollkästchen neben der Funktion, die Sie im Abschnitt [Schritt 4](#) (zum Beispiel **tutorial-lambda-convert**) erstellt haben.

4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie im Bestätigungsdiaologfeld Delete (Löschen) die Option Delete (Löschen) aus.

Löschen Sie die CloudWatch Protokollgruppe

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im linken Navigationsbereich Protokolle und dann Protokollgruppen aus.
3. Aktivieren Sie das Kontrollkästchen neben der Protokollgruppe, deren Name mit der Lambda-Funktion endet, die Sie im Abschnitt [Schritt 4](#) (zum Beispiel **tutorial-lambda-convert**) erstellt haben.
4. Wählen Sie Actions (Aktionen) und dann Delete log group(s) (Protokollgruppe(n) löschen) aus.
5. Wählen Sie im Dialogfeld Delete log group(s) (Protokollgruppe(n) löschen) die Option Delete (Löschen) aus.

Löschen Sie die IAM Rollen zusammen mit den Inline-Richtlinien für die IAM Rollen

Gehen Sie wie folgt vor, um die IAM Rollen zu löschen, die Sie in [Schritt 2](#), [Schritt 3](#) und [Schritt 6](#) erstellt haben:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen), und aktivieren Sie dann die Kontrollkästchen neben den Rollennamen, die Sie löschen möchten.
3. Wählen Sie oben auf der Seite Delete (Löschen) aus.
4. Geben Sie in das Bestätigungsfeld die erforderliche Antwort basierend auf der Eingabeaufforderung in das Texteingabefeld ein, und wählen Sie dann Löschen aus.

Löschen Sie die vom Kunden verwaltete Richtlinie IAM

Gehen Sie wie folgt vor, um die vom Kunden verwaltete IAM Richtlinie zu löschen, die Sie in [Schritt 6](#) erstellt haben:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.

3. Wählen Sie das Optionsfeld neben der Richtlinie aus, die Sie in [Schritt 6](#) (zum Beispiel **tutorial-s3batch-policy**) erstellt haben. Sie können über das Suchfeld die Liste der Richtlinien filtern.
4. Wählen Sie Aktionen und anschließend Löschen.
5. Bestätigen Sie, dass Sie diese Richtlinie löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Leeren Sie die S3-Buckets

Um die S3-Buckets zu leeren, die Sie in [Voraussetzungen](#), [Schritt 1](#) und [Schritt 5](#) erstellt haben, tun Sie Folgendes:

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie leeren möchten. Wählen Sie anschließend Empty (Leeren) aus.
4. Bestätigen Sie auf der Seite Empty bucket (Bucket leeren), dass Sie den Bucket leeren möchten, indem Sie **permanently delete** in das Textfeld eingeben und dann Empty (Leeren) auswählen.

Löschen der S3-Buckets

Um die S3-Buckets zu löschen, die Sie in [Voraussetzungen](#), [Schritt 1](#) und [Schritt 5](#) erstellt haben, tun Sie Folgendes:

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie löschen möchten.
4. Wählen Sie Löschen.
5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Nächste Schritte

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie weitere relevante Anwendungsfälle untersuchen:

- Sie können Amazon verwenden CloudFront , um die transkodierte Mediendateien an Zuschauer auf der ganzen Welt zu streamen. Weitere Informationen finden Sie unter [Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53](#).
- Sie können Videos in dem Moment transcodieren, indem Sie sie in den S3-Quell-Bucket hochladen. Zu diesem Zweck können Sie einen Amazon S3 S3-Event-Trigger konfigurieren, der automatisch die Lambda-Funktion aufruft, mit der neue Objekte in S3 transkodiert werden. MediaConvert Weitere Informationen finden Sie unter [Tutorial: Verwenden eines Amazon S3 S3-Triggers zum Aufrufen einer Lambda-Funktion](#) in AWS Lambda Leitfaden für Entwickler.

Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität

Mit dem mehrteiligen Upload können Sie ein einzelnes Objekt als Satz aus mehreren Teilen hochladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt. Wenn Ihre Objektgröße 100 MB erreicht, sollten Sie in der Regel mehrteilige Uploads verwenden, anstatt das Objekt in einem einzigen Vorgang hochzuladen. Weitere Informationen zu mehrteiligen Uploads finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#) Einschränkungen im Zusammenhang mit mehrteiligen Uploads finden Sie unter [Einschränkungen mehrteiliger Amazon-S3-Uploads](#)

Sie können Prüfsummen verwenden, um sicherzustellen, dass Assets beim Kopieren nicht verändert werden. Das Durchführen einer Prüfsumme besteht darin, mithilfe eines Algorithmus sequentiell über jedes Byte in einer Datei zu iterieren. Amazon S3 bietet mehrere Prüfsummenoptionen zur Überprüfung der Integrität von Daten. Wir empfehlen Ihnen, diese Integritätsprüfungen als bewährte Methode für die Haltbarkeit durchzuführen und um sicherzustellen, dass jedes Byte unverändert übertragen wird. Amazon S3 unterstützt auch die folgenden Algorithmen: SHA -1, SHA -256CRC32, undCRC32C. Amazon S3 verwendet einen oder mehrere dieser Algorithmen, um einen zusätzlichen

Prüfsummenwert zu berechnen und ihn als Teil der Objektmetadaten zu speichern. Weitere Informationen zu Prüfsummen finden Sie unter [Überprüfung der Objektintegrität](#).

Ziel

In diesem Tutorial erfahren Sie, wie Sie ein Objekt mithilfe eines mehrteiligen Uploads und einer zusätzlichen SHA -256-Prüfsumme auf Amazon S3 hochladen. AWS Befehlszeilenschnittstelle (AWS CLI). Sie erfahren auch, wie Sie die Datenintegrität des Objekts überprüfen können, indem Sie den MD5 Hash und die SHA -256-Prüfsumme des hochgeladenen Objekts berechnen.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie eine große Datei](#)
- [Schritt 2: Teilen Sie die Datei in mehrere Dateien auf](#)
- [Schritt 3: Erstellen Sie den mehrteiligen Upload mit einer zusätzlichen Prüfsumme](#)
- [Schritt 4: Laden Sie die Teile Ihres mehrteiligen Uploads hoch](#)
- [Schritt 5: Listet alle Teile eures mehrteiligen Uploads auf](#)
- [Schritt 6: Schließen Sie den mehrteiligen Upload ab](#)
- [Schritt 7: Vergewissern Sie sich, dass das Objekt in Ihren Bucket hochgeladen wurde](#)
- [Schritt 8: Überprüfen Sie die Objektintegrität mit einer MD5 Prüfsumme](#)
- [Schritt 9: Überprüfen Sie die Objektintegrität mit einer zusätzlichen Prüfsumme](#)
- [Schritt 10: Bereinigen Ihrer Ressourcen](#)

Voraussetzungen

- Bevor Sie mit diesem Tutorial beginnen, stellen Sie sicher, dass Sie Zugriff auf einen Amazon S3 S3-Bucket haben, in den Sie hochladen können. Weitere Informationen finden Sie unter [Erstellen eines Buckets](#).
- Sie müssen die haben AWS CLI installiert und konfiguriert. Wenn Sie das nicht haben AWS CLI installiert, siehe [Installation oder Aktualisierung auf die neueste Version von AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.
- Alternativ können Sie ausführen AWS CLIBefehle von der Konsole aus mit AWS CloudShell. AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der AWS Management Console. Weitere Informationen finden Sie unter [Was ist CloudShell?](#) und [Erste Schritte mit AWS CloudShell](#) in der AWS CloudShell Benutzerleitfaden.

Schritt 1: Erstellen Sie eine große Datei

Wenn Sie bereits eine Datei zum Hochladen bereit haben, können Sie die Datei für dieses Tutorial verwenden. Andernfalls erstellen Sie mit den folgenden Schritten eine 15-MB-Datei. Informationen zu Beschränkungen im Zusammenhang mit mehrteiligen Uploads finden Sie unter [Einschränkungen mehrteiliger Amazon-S3-Uploads](#)

Um eine große Datei zu erstellen

Verwenden Sie einen der folgenden Befehle, um Ihre Datei zu erstellen, je nachdem, welches Betriebssystem Sie verwenden.

Linux oder macOS

Um eine 15-MB-Datei zu erstellen, öffnen Sie Ihr lokales Terminal und führen Sie den folgenden Befehl aus:

```
dd if=/dev/urandom of=census-data.bin bs=1M count=15
```

Dieser Befehl erstellt eine mit zufälligen Bytes `census-data.bin` gefüllte Datei mit einer Größe von 15 MB.

Windows

Um eine 15-MB-Datei zu erstellen, öffnen Sie Ihr lokales Terminal und führen Sie den folgenden Befehl aus:

```
fsutil file createnew census-data.bin 15728640
```

Mit diesem Befehl wird eine Datei `census-data.bin` mit einem Namen und einer Größe von 15 MB mit beliebigen Daten (15728640 Byte) erstellt.

Schritt 2: Teilen Sie die Datei in mehrere Dateien auf

Um den mehrteiligen Upload durchzuführen, müssen Sie Ihre große Datei in kleinere Teile aufteilen. Sie können dann die kleineren Teile hochladen, indem Sie den mehrteiligen Upload-Prozess verwenden. In diesem Schritt wird gezeigt, wie Sie die in [Schritt 1](#) erstellte große Datei in kleinere Teile aufteilen. Im folgenden Beispiel wird eine 15-MB-Datei mit dem Namen `verwendetcensus-data.bin`.

Um eine große Datei in Teile aufzuteilen

Linux oder macOS

Verwenden Sie den `split` Befehl, um die große Datei in 5 MB-Teile zu unterteilen. Öffnen Sie Ihr Terminal und führen Sie Folgendes aus:

```
split -b 5M -d census-data.bin census-part
```

Dieser Befehl teilt sich `census-data.bin` in 5 MB große Teile mit dem Namen `census-part**`, wobei `**` ein numerisches Suffix beginnt. `00`

Windows

Um die große Datei aufzuteilen, verwenden Sie PowerShell. Öffnen Sie [PowerShell](#) und führen Sie das folgende Skript aus:

```
$inputFile = "census-data.bin"
$outputFilePrefix = "census-part"
$chunkSize = 5MB

$fs = [System.IO.File]::OpenRead($inputFile)
$buffer = New-Object byte[] $chunkSize
$fileNumber = 0

while ($fs.Position -lt $fs.Length) {
    $bytesRead = $fs.Read($buffer, 0, $chunkSize)
    $outputFile = "{0}{1:D2}" -f $outputFilePrefix, $fileNumber
    $fileStream = [System.IO.File]::Create($outputFile)
    $fileStream.Write($buffer, 0, $bytesRead)
    $fileStream.Close()
    $fileNumber++
}

$fs.Close()
```

Dieses PowerShell Skript liest die große Datei in Blöcken von 5 MB und schreibt jeden Abschnitt in eine neue Datei mit einem numerischen Suffix.

Nachdem Sie den entsprechenden Befehl ausgeführt haben, sollten Sie die Teile in dem Verzeichnis sehen, in dem Sie den Befehl ausgeführt haben. Jeder Teil hat ein Suffix, das seiner Artikelnummer entspricht, zum Beispiel:

```
census-part00 census-part01 census-part02
```

Schritt 3: Erstellen Sie den mehrteiligen Upload mit einer zusätzlichen Prüfsumme

Um den mehrteiligen Upload-Vorgang zu starten, müssen Sie die mehrteilige Upload-Anfrage erstellen. In diesem Schritt wird der mehrteilige Upload initiiert und eine zusätzliche Prüfsumme für die Datenintegrität angegeben. Im folgenden Beispiel wird die Prüfsumme -256 verwendet. SHA Wenn Sie Metadaten angeben möchten, die das hochgeladene Objekt beschreiben, müssen Sie diese in der Anforderung angeben, um den mehrteiligen Upload zu initiieren.

Note

In diesem Schritt und den nachfolgenden Schritten verwendet dieses Tutorial den zusätzlichen Algorithmus SHA -256. Sie können optional eine weitere zusätzliche Prüfsumme für diese Schritte verwenden, z. B. CRC32CRC32C, oder SHA -1. Wenn Sie einen anderen Algorithmus verwenden, müssen Sie ihn in allen Schritten des Tutorials verwenden.

Um den mehrteiligen Upload zu starten

Verwenden Sie in Ihrem Terminal den folgenden `create-multipart-upload` Befehl, um einen mehrteiligen Upload für Ihren Bucket zu starten. Ersetzen Sie es `amzn-s3-demo-bucket1` durch Ihren tatsächlichen Bucket-Namen. Ersetzen Sie außerdem den `census_data_file` durch den von Ihnen gewählten Dateinamen. Dieser Dateiname wird zum Objektschlüssel, wenn der Upload abgeschlossen ist.

```
aws s3api create-multipart-upload --bucket amzn-s3-demo-bucket1 --key  
'census_data_file' --checksum-algorithm sha256
```

Wenn Ihre Anfrage erfolgreich ist, erhalten Sie eine JSON Ausgabe wie die folgende:

```
{  
  "ServerSideEncryption": "AES256",  
  "ChecksumAlgorithm": "SHA256",  
  "Bucket": "amzn-s3-demo-bucket1",  
  "Key": "census_data_file",
```

```
"UploadId":  
"cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sN0KPeMhpKQRrbT9k0ePmgo0TCj9K83T4e2Gb5hQvNoNpCKqyb8m3"  
}
```

Note

Wenn Sie eine Anfrage zum Initiieren eines mehrteiligen Uploads senden, gibt Amazon S3 eine Antwort mit einer Upload-ID zurück, als eindeutige Kennung für den Multipart-Upload. Sie müssen diese Upload-ID immer angeben, wenn Sie Teile hochladen, die Teile auflisten, einen Upload abschließen oder einen Upload abbrechen. Sie müssen die Bucket Werte `UploadId`, und für spätere Schritte verwenden `Key`, also stellen Sie sicher, dass Sie diese speichern.

Wenn Sie den mehrteiligen Upload mit zusätzlichen Prüfsummen verwenden, müssen die Artikelnummern außerdem fortlaufend sein. Wenn Sie nicht aufeinander folgende Artikelnummern verwenden, kann die `complete-multipart-upload` Anfrage zu einer `HTTP 500 Internal Server Error` führen.

Schritt 4: Laden Sie die Teile Ihres mehrteiligen Uploads hoch

In diesem Schritt laden Sie die Teile Ihres mehrteiligen Uploads in Ihren S3-Bucket hoch. Verwenden Sie den `upload-part` Befehl, um jeden Teil einzeln hochzuladen. Dieser Vorgang erfordert die Angabe der Upload-ID, der Artikelnummer und der hochzuladenden Datei für jedes Teil.

Um die Teile hochzuladen

1. Wenn Sie ein Teil hochladen, müssen Sie zusätzlich zur Upload-ID eine Artikelnummer mithilfe des `--part-number` Arguments angeben. Sie können jede Teilenummer zwischen 1 und 10.000 wählen. Die Teilenummer identifiziert eindeutig einen Teil und seine Position im Objekt, das Sie hochladen. Die von Ihnen gewählte Artikelnummer muss in einer aufeinanderfolgenden Reihenfolge angegeben werden (sie kann beispielsweise 1, 2 oder 3 sein). Wenn Sie einen neuen Teil mit derselben Teilenummer hochladen wie bereits einmal zuvor, wird der früher hochgeladene Teil überschrieben.
2. Verwenden Sie den `upload-part` Befehl, um jeden Teil Ihres mehrteiligen Uploads hochzuladen. Der `--upload-id` ist derselbe wie in der Ausgabe, die mit dem `create-multipart-upload` Befehl in [Schritt 3](#) erstellt wurde. Verwenden Sie den folgenden Befehl, um den ersten Teil Ihrer Daten hochzuladen:

```
aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key
'census_data_file' --part-number 1 --body census-part00 --upload-id
"cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sN0KPeMhpKQRrbT9k0ePmgo0TCj9K83T4e2Gb5hQvNoNpCKqy"
--checksum-algorithm SHA256
```

Nach Abschluss jedes `upload-part` Befehls sollten Sie eine Ausgabe wie das folgende Beispiel sehen:

```
{
  "ServerSideEncryption": "AES256",
  "ETag": "\"e611693805e812ef37f96c9937605e69\"",
  "ChecksumSHA256": "QL18R4i4+SaJlrl8Zlcutc5TbZtw2NwB81TXkd3GH0="
}
```

3. Für nachfolgende Teile erhöhen Sie die Artikelnummer entsprechend:

```
aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --
part-number <part-number> --body <file-path> --upload-id "<your-upload-id>" --
checksum-algorithm SHA256
```

Verwenden Sie beispielsweise den folgenden Befehl, um den zweiten Teil hochzuladen:

```
aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key
'census_data_file' --part-number 2 --body census-part01 --upload-id
"cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sN0KPeMhpKQRrbT9k0ePmgo0TCj9K83T4e2Gb5hQvNoNpCKqy"
--checksum-algorithm SHA256
```

Amazon S3 gibt ein Entitäts-Tag (ETag) und zusätzliche Prüfsummen für jeden hochgeladenen Teil als Header in der Antwort zurück.

4. Verwenden Sie den `upload-part` Befehl weiter, bis Sie alle Teile Ihres Objekts hochgeladen haben.

Schritt 5: Listet alle Teile eures mehrteiligen Uploads auf

Um den mehrteiligen Upload abzuschließen, benötigen Sie eine Liste aller Teile, die für diesen speziellen mehrteiligen Upload hochgeladen wurden. Die Ausgabe des `list-parts` Befehls

enthält Informationen wie Bucket-Namen, Schlüssel, Upload-ID, ArtikelnummereTag, zusätzliche Prüfsummen und mehr. Es ist hilfreich, diese Ausgabe in einer Datei zu speichern, sodass Sie sie für den nächsten Schritt verwenden können, wenn Sie den mehrteiligen Upload-Vorgang abschließen. Sie können eine JSON Ausgabedatei erstellen, die mit `parts.json` der folgenden Methode aufgerufen wird.

Um eine Datei zu erstellen, die alle Teile auflistet

1. Verwenden Sie den folgenden `list-parts` Befehl, um eine JSON Datei mit den Details aller hochgeladenen Teile zu generieren. `amzn-s3-demo-bucket1` Ersetzen Sie es durch Ihren tatsächlichen Bucket-Namen und `<your-upload-id>` durch die Upload-ID, die Sie in [Schritt 3](#) erhalten haben. Weitere Informationen zu dem `list-parts` Befehl finden Sie unter [list-parts](#) in der AWS Command Line Interface Benutzerleitfaden.

```
aws s3api list-parts --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --
upload-id <your-upload-id> --query '{Parts: Parts[*].{PartNumber: PartNumber, ETag:
ETag, ChecksumSHA256: ChecksumSHA256}}' --output json > parts.json
```

Eine neue Datei namens `parts.json` wird generiert. Die Datei enthält die JSON formatierten Informationen für alle Ihre hochgeladenen Teile. Die `parts.json` Datei enthält wichtige Informationen für jeden Teil Ihres mehrteiligen Uploads, wie z. B. die Artikelnummern und die entsprechenden ETag Werte, die für den Abschluss des mehrteiligen Uploads erforderlich sind.

2. Öffnen Sie `parts.json` sie mit einem beliebigen Texteditor oder über das Terminal. Hier ist die Beispielausgabe:

```
{
  "Parts": [
    {
      "PartNumber": 1,
      "ETag": "\"3c3097f89e2a2fece47ac54b243c9d97\"",
      "ChecksumSHA256": "fTPVHfyNHdv5Vkr4S3EewdyioXECv7JBxN+d4FXYYTw="
    },
    {
      "PartNumber": 2,
      "ETag": "\"03c71cc160261b20ab74f6d2c476b450\"",
      "ChecksumSHA256": "VDWTa8enj0vULBA03W2a6C+5/7ZnNjrnLAPA1QVc3FE="
    },
    {
      "PartNumber": 3,
```

```
        "ETag": "\"81ae0937404429a97967dffa7eb4affb\"",
        "ChecksumSHA256": "cVvkXehUlzcwrBrXgPIM+EKQXPuvWist8mLUTCs4bg8="
    }
]
}
```

Schritt 6: Schließen Sie den mehrteiligen Upload ab

Nachdem Sie alle Teile Ihres mehrteiligen Uploads hochgeladen und aufgelistet haben, besteht der letzte Schritt darin, den mehrteiligen Upload abzuschließen. In diesem Schritt werden alle hochgeladenen Teile zu einem einzigen Objekt in Ihrem S3-Bucket zusammengeführt.

Note

Sie können die Objekt-Prüfsumme vor dem Aufruf berechnen, `complete-multipart-upload` indem Sie sie `--checksum-sha256` in Ihre Anfrage aufnehmen. Wenn die Prüfsummen nicht übereinstimmen, schlägt Amazon S3 die Anfrage fehl. Weitere Informationen finden Sie unter [complete-multipart-upload](#) in der AWS Command Line Interface Benutzerhandbuch.

Um den mehrteiligen Upload abzuschließen

Verwenden Sie den Befehl, um den mehrteiligen Upload abzuschließen. `complete-multipart-upload` Für diesen Befehl sind die in [Schritt 5](#) erstellte `parts.json` Datei, Ihr Bucket-Name und die Upload-ID erforderlich. `<amzn-s3-demo-bucket1>` Ersetzen Sie durch Ihren Bucket-Namen und `<your-upload-id>` die Upload-ID von `parts.json`.

```
aws s3api complete-multipart-upload --multipart-upload file://parts.json --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --upload-id <your-upload-id>
```

Hier ist die Beispielausgabe:

```
{
  "ServerSideEncryption": "AES256",
  "Location": "https://amzn-s3-demo-bucket1.s3.us-east-2.amazonaws.com/census_data_file",
}
```

```
"Bucket": "amzn-s3-demo-bucket1",
"Key": "census_data_file",
"ETag": "\"f453c6dcca969c457efdf9b1361e291-3\"",
"ChecksumSHA256": "aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=-3"
}
```

Note

Löschen Sie die einzelnen Teildateien noch nicht. Sie benötigen die einzelnen Teile, damit Sie Prüfsummen auf sie anwenden können, um die Integrität des zusammengeführten Objekts zu überprüfen.

Schritt 7: Vergewissern Sie sich, dass das Objekt in Ihren Bucket hochgeladen wurde

Nach Abschluss des mehrteiligen Uploads können Sie überprüfen, ob das Objekt erfolgreich in Ihren S3-Bucket hochgeladen wurde. Verwenden Sie den Befehl, um die Objekte in Ihrem Bucket aufzulisten und das Vorhandensein Ihrer neu hochgeladenen Datei zu bestätigen `list-objects-v2`

Um das hochgeladene Objekt aufzulisten

Um die Objekte in Ihrem Bucket aufzulisten, verwenden Sie den `list-objects-v2` Befehl `bucket.amzn-s3-demo-bucket1`. Ersetzen Sie es durch Ihren tatsächlichen Bucket-Namen:

```
aws s3api list-objects-v2 --bucket amzn-s3-demo-bucket1
```

Dieser Befehl gibt eine Liste der Objekte in Ihrem Bucket zurück. Suchen Sie in der Objektliste nach Ihrer hochgeladenen Datei (zum Beispiel `census_data_file`).

Weitere Informationen finden Sie im Abschnitt [Beispiele](#) für den `list-objects-v2` Befehl in der AWS Command Line Interface Benutzerleitfaden.

Schritt 8: Überprüfen Sie die Objektintegrität mit einer MD5 Prüfsumme

Wenn Sie ein Objekt hochladen, können Sie einen Prüfsummenalgorithmus angeben, den Amazon S3 verwenden soll. Standardmäßig speichert Amazon S3 den MD5 Byte-Digest als den des ETag

Objekts. Bei mehrteiligen Uploads ETag handelt es sich nicht um die Prüfsumme für das gesamte Objekt, sondern um eine Zusammenstellung von Prüfsummen für jeden einzelnen Teil.

Um die Objektintegrität mithilfe einer Prüfsumme zu überprüfen MD5

1. Führen Sie eine `head-object` Anfrage ETag aus, um das hochgeladene Objekt abzurufen:

```
aws s3api head-object --bucket amzn-s3-demo-bucket1 --key census_data_file
```

Hier ist die Beispielausgabe:

```
{
  "AcceptRanges": "bytes",
  "LastModified": "2024-07-26T19:04:13+00:00",
  "ContentLength": 16106127360,
  "ETag": "\"f453c6dcccc969c457efdf9b1361e291-3\"",
  "ContentType": "binary/octet-stream",
  "ServerSideEncryption": "AES256",
  "Metadata": {}
}
```

An das Ende dieses Etags ist „-3“ angehängt. Dies weist darauf hin, dass das Objekt in drei Teilen mithilfe eines mehrteiligen Uploads hochgeladen wurde.

2. Als Nächstes berechnen Sie die MD5 Prüfsumme der einzelnen Teile mit dem `md5sum` Befehl. Stellen Sie sicher, dass Sie den richtigen Pfad zu Ihren Bauteildateien angeben:

```
md5sum census-part*
```

Hier ist die Beispielausgabe:

```
e611693805e812ef37f96c9937605e69 census-part00
63d2d5da159178785bfd6b6a5c635854 census-part01
95b87c7db852451bb38b3b44a4e6d310 census-part02
```

3. Kombinieren Sie für diesen Schritt die MD5 Hashes manuell zu einer Zeichenfolge. Führen Sie dann den folgenden Befehl aus, um die Zeichenfolge in eine Binärdatei zu konvertieren und die MD5 Prüfsumme des Binärwerts zu berechnen:

```
echo  
"e611693805e812ef37f96c9937605e6963d2d5da159178785bfd6b6a5c63585495b87c7db852451bb38b3b44a"  
| xxd -r -p | md5sum
```

Hier ist die Beispielausgabe:

```
f453c6dcca969c457efdf9b1361e291 -
```

Dieser Hashwert sollte mit dem Hashwert des ursprünglichen ETag Werts in [Schritt 1](#) übereinstimmen, der die Integrität des `census_data_file` Objekts bestätigt.

Wenn Sie Amazon S3 anweisen, zusätzliche Prüfsummen zu verwenden, berechnet Amazon S3 den Prüfsummenwert für jeden Teil und speichert die Werte. Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie Folgendes verwenden. `list-parts`

Weitere Informationen zur Funktionsweise von Prüfsummen mit mehrteiligen Objekten finden Sie unter [Überprüfung der Objektintegrität](#).

Schritt 9: Überprüfen Sie die Objektintegrität mit einer zusätzlichen Prüfsumme

In diesem Schritt verwendet dieses Tutorial SHA -256 als zusätzliche Prüfsumme, um die Objektintegrität zu überprüfen. Wenn Sie eine andere zusätzliche Prüfsumme verwendet haben, verwenden Sie stattdessen diesen Prüfsummenwert.

Um die Objektintegrität zu überprüfen mit SHA256

1. Führen Sie den folgenden Befehl in Ihrem Terminal aus, einschließlich des `--checksum-mode enabled` Arguments, um den ChecksumSHA256 Wert Ihres Objekts anzuzeigen:

```
aws s3api head-object --bucket amzn-s3-demo-bucket1 --key census_data_file --  
checksum-mode enabled
```

Hier ist die Beispielausgabe:

```
{
  "AcceptRanges": "bytes",
  "LastModified": "2024-07-26T19:04:13+00:00",
  "ContentLength": 16106127360,
  "ChecksumSHA256": "aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=-3",
  "ETag": "\"f453c6dcca969c457efdf9b1361e291-3\"",
  "ContentType": "binary/octet-stream",
  "ServerSideEncryption": "AES256",
  "Metadata": {}
}
```

2. Verwenden Sie die folgenden Befehle, um die ChecksumSHA256 Werte für die einzelnen Teile in Base64 zu dekodieren und sie in einer Binärdatei namens `outfile` zu speichern. Diese Werte finden Sie in Ihrer `parts.json` Datei. Ersetzen Sie die Base64-Beispiel-Strings durch Ihre tatsächlichen ChecksumSHA256 Werte.

```
echo "QL18R4i4+SaJlrl8Zlcutc5TbZtw2NwB8lTXkd3GH0=" | base64 --decode >> outfile
echo "xCdgs1K5Bm4jWETyw/CmGYr+m602DcGfpckx5NVokvE=" | base64 --decode >> outfile
echo "f5wsfsa5bB+yXuwzqG1Bst91uYneqGD3CCidpb54mAo=" | base64 --decode >> outfile
```

3. Führen Sie den folgenden Befehl aus, um die SHA256 Prüfsumme von `outfile` zu berechnen:

```
sha256sum outfile
```

Hier ist die Beispielausgabe:

```
688f04a24b42768b6353c06ae3a0eb3c2c50086b8670f221279d67a16b3a86f9 outfile
```

Nehmen Sie im nächsten Schritt den Hashwert und konvertieren Sie ihn in einen Binärwert. Dieser Binärwert sollte mit dem ChecksumSHA256 Wert aus [Schritt 1](#) übereinstimmen.

4. [Wandeln Sie die SHA256 Prüfsumme aus Schritt 3 in eine Binärdatei um und codieren Sie sie dann nach Base64, um zu überprüfen, ob sie mit dem ChecksumSHA256 Wert aus Schritt 1 übereinstimmt:](#)

```
echo "688f04a24b42768b6353c06ae3a0eb3c2c50086b8670f221279d67a16b3a86f9" | xxd -r -p  
| base64
```

Hier ist die Beispielausgabe:

```
aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=
```

Diese Ausgabe sollte bestätigen, dass die Base64-Ausgabe mit dem ChecksumSHA256 Wert aus der `head-object` Befehlsausgabe übereinstimmt. Wenn die Ausgabe mit dem Prüfsummenwert übereinstimmt, ist das Objekt gültig.

Important

- Wenn Sie Amazon S3 anweisen, zusätzliche Prüfsummen zu verwenden, berechnet Amazon S3 die Prüfsummenwerte für jeden Teil und speichert diese Werte.
- Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie den Befehl verwenden. `list-parts`

Schritt 10: Bereinigen Ihrer Ressourcen

Wenn Sie die in diesem Tutorial erstellten Dateien bereinigen möchten, verwenden Sie die folgende Methode. Anweisungen zum Löschen der in Ihren S3-Bucket hochgeladenen Dateien finden Sie unter [Löschen von Amazon-S3-Objekten](#).

Löschen Sie die in [Schritt 1](#) erstellten lokalen Dateien:

Um die Dateien zu entfernen, die Sie für Ihren mehrteiligen Upload erstellt haben, führen Sie den folgenden Befehl in Ihrem Arbeitsverzeichnis aus:

```
rm census-data.bin census-part* outfile parts.json
```

Tutorial: Konfigurieren einer statischen Website auf Amazon S3

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Sie können einen Amazon S3-Bucket so konfigurieren, dass er sich wie eine Website verhält. In diesem Beispiel sehen Sie die Schritte für das Hosten einer Website auf Amazon S3.

Important

Für das folgende Tutorial ist die Deaktivierung des öffentlichen Zugriffs erforderlich. Wir empfehlen, alle Einstellungen für das Blockieren des öffentlichen Zugriffs aktiviert zu lassen. Wenn Sie alle vier Block Public Access-Einstellungen aktiviert lassen und eine statische Website hosten möchten, können Sie Amazon CloudFront Origin Access Control (OAC) verwenden. Amazon CloudFront bietet die Funktionen, die für die Einrichtung einer sicheren statischen Website erforderlich sind. Statische Amazon S3 S3-Websites unterstützen nur HTTP Endpunkte. Amazon CloudFront verwendet den dauerhaften Speicher von Amazon S3 und bietet gleichzeitig zusätzliche Sicherheitsheader, wie HTTPS z. HTTPS erhöht die Sicherheit, indem es eine normale HTTP Anfrage verschlüsselt und vor häufigen Cyberangriffen schützt. Weitere Informationen finden Sie unter [Erste Schritte mit einer sicheren statischen Website](#) im Amazon CloudFront Developer Guide.

Themen

- [Schritt 1: Erstellen eines Buckets](#)
- [Schritt 2: Aktivieren des statischen Website-Hostings](#)
- [Schritt 3: Bearbeiten der Block-Public-Access-Einstellungen](#)

- [Schritt 4: Hinzufügen einer Bucket-Richtlinie, die den Inhalt Ihres Buckets öffentlich verfügbar macht](#)
- [Schritt 5: Konfigurieren eines Indextdokuments](#)
- [Schritt 6: Konfigurieren eines Fehlerdokuments](#)
- [Schritt 7: Testen des Website-Endpunkts](#)
- [Schritt 8: Bereinigen](#)

Schritt 1: Erstellen eines Buckets

Die folgenden Anweisungen geben einen Überblick darüber, wie Sie Ihre Buckets für das Website-Hosting erstellen. Eine ausführliche step-by-step Anleitung zur Erstellung eines Buckets finden Sie unter [Erstellen eines Buckets](#).

So erstellen Sie einen Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Create Bucket (Bucket erstellen).
3. Geben Sie den Bucket-Namen ein (z. B. **example.com**).
4. Wählen Sie die Region aus, in der Sie Ihren Bucket erstellen möchten.

Wählen Sie eine Region in Ihrer Nähe aus, um Latenz und Kosten auf einem Minimum zu halten oder behördliche Vorschriften zu erfüllen. Die von Ihnen ausgewählte Region bestimmt Ihren Amazon-S3-Website-Endpunkt. Weitere Informationen finden Sie unter [Website-Endpunkte](#).

5. Um die Standardeinstellungen zu übernehmen und den Bucket zu erstellen, wählen Sie Create (Erstellen).

Schritt 2: Aktivieren des statischen Website-Hostings

Nach der Erstellung eines Buckets können Sie das statische Website-Hosting für Ihren Bucket aktivieren. Sie können einen neuen Bucket erstellen oder einen vorhandenen Bucket verwenden.

So aktivieren Sie das statische Website-Hosting

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie das Hosting statischer Websites aktivieren wollen.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Wählen Sie Use this bucket to host a website (Diesen Bucket zum Hosten einer Website verwenden).
6. Wählen Sie unter Static website hosting (Hosting statischer Websites) die Option Enable (Aktivieren) aus
7. Geben Sie unter Index document (Index-Dokument) den Dateinamen des Index-Dokuments ein, der typischerweise `index.html` ist.

Beim Namen des Indextdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss exakt mit dem Dateinamen des HTML Indextdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie Ihren Bucket für das Hosting von Websites konfigurieren, müssen Sie ein Indextdokument angeben. Amazon S3 gibt dieses Indextdokument zurück, wenn Anfragen an die Root-Domäne oder einen der Unterordner gestellt werden. Weitere Informationen finden Sie unter [Konfigurieren eines Indextdokuments](#).

8. Um ein eigenes benutzerdefiniertes Fehlerdokument für Fehler der Klasse 4XX bereitzustellen, geben Sie unter Fehlerdokument den Dateinamen des benutzerdefinierten Fehlerdokuments ein.

Beim Namen des Fehlerdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss genau mit dem Dateinamen des HTML Fehlerdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie kein benutzerdefiniertes Fehlerdokument angeben und ein Fehler auftritt, gibt Amazon S3 ein HTML Standard-Fehlerdokument zurück. Weitere Informationen finden Sie unter [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#).

9. (Optional) Wenn Sie erweiterte Umleitungsregeln angeben möchten, geben Sie im Feld JSONUmleitungsregeln eine Beschreibung der Regeln ein.

Beispielsweise können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten. Weitere Informationen finden Sie unter [Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen](#).

10. Wählen Sie Save Changes (Änderungen speichern).

Amazon S3 ermöglicht statisches Website-Hosting für Ihren Bucket. Unten auf der Seite sehen Sie unter Static website hosting (Hosting statischer Websites) den Website-Endpoint für Ihren Bucket.

11. Notieren Sie unter Static website hosting (Statisches Website-Hosting) den Wert für Endpoint (Endpoint).

Der Endpoint (Endpoint) ist der Amazon-S3-Website-Endpoint für Ihren Bucket. Nachdem Sie den Bucket als statische Website konfiguriert haben, können Sie diesen Endpoint verwenden, um Ihre Website zu testen.

Schritt 3: Bearbeiten der Block-Public-Access-Einstellungen


Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

Warning

Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#) dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugriffs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#) 



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

- Block *all* public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

 - Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
 - Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.
 - Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
 - Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die [Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten](#), bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Schritt 4: Hinzufügen einer Bucket-Richtlinie, die den Inhalt Ihres Buckets öffentlich verfügbar macht

Nachdem Sie die Einstellungen für S3 Block Public Access bearbeitet haben, können Sie eine Bucket-Richtlinie hinzufügen, um öffentlichen Lesezugriff auf den Bucket zu gewähren. Wenn Sie öffentlichen Lesezugriff gewähren, kann jeder im Internet auf Ihren Bucket zugreifen.

⚠ Important

Die zuvor genannte Richtlinie ist nur ein Beispiel und erlaubt Vollzugriff auf die Inhalte Ihres Buckets. Bevor Sie mit diesem Schritt fortfahren, lesen Sie den Abschnitt [Wie kann ich die Dateien in meinem Amazon-S3-Bucket sichern?](#), um sicherzustellen, dass Sie die bewährten Methoden zum Sichern der Dateien in Ihrem S3-Bucket und die Risiken in Zusammenhang mit der Gewährung von öffentlichem Zugriff kennen.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Bucket Policy (Bucket-Richtlinie) Edit (Bearbeiten).
4. Um öffentlichen Lesezugriff auf Ihre Website zu gewähren, kopieren Sie die folgende Bucket-Richtlinie und fügen Sie sie in den Bucket policy editor (Bucket-Richtlinieneditor) ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::Bucket-Name/*"
      ]
    }
  ]
}
```

5. Aktualisieren Sie den Resource zu Ihrem Bucket-Namen.

Im obigen Beispiel für eine Bucket-Richtlinie *Bucket-Name* ist ein Platzhalter für den Bucket-Namen. Um diese Bucket-Richtlinie mit Ihrem eigenen Bucket zu verwenden, müssen Sie diesen Namen so aktualisieren, dass er mit Ihrem Bucket übereinstimmt.

6. Wählen Sie Save Changes (Änderungen speichern).

Es wird eine Meldung angezeigt, die darauf hinweist, dass die Bucket-Richtlinie erfolgreich hinzugefügt wurde.

Wenn die Fehlermeldung `Policy has invalid resource` angezeigt wird, bestätigen Sie, dass der Bucket-Name in der Bucket-Richtlinie mit Ihrem Bucket-Namen übereinstimmt. Informationen zum Hinzufügen einer Bucket-Richtlinie finden Sie unter [Wie füge ich eine S3-Bucket-Richtlinie hinzu?](#)

Wenn Sie eine Fehlermeldung erhalten und die Bucket-Richtlinie nicht speichern können, überprüfen Sie Ihr Konto und die Bucket-Einstellungen für Block Public Access, um zu bestätigen, dass Sie den öffentlichen Zugriff auf den Bucket zulassen.

Schritt 5: Konfigurieren eines Indextdokuments

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indextdokuments ein (z. B. **index.html**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Indextdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie das Indextdokument

1. Erstellen Sie eine Datei `index.html`.

Wenn Sie noch keine `index.html` Datei haben, können Sie wie folgt HTML eine Datei erstellen:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>My Website Home Page</title>
</head>
<body>
  <h1>Welcome to my website</h1>
  <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

2. Speichern Sie die Indexdatei lokal.

Der Dateiname des Indextdokuments muss genau mit dem Namen des Indextdokuments übereinstimmen, den Sie im Dialogfeld Static website hosting (Statisches Website-Hosting)

eingeben. Beim Namen des Indexdokuments wird die Groß- und Kleinschreibung berücksichtigt. Wenn Sie beispielsweise im Dialogfeld Static website hosting (Statisches Website-Hosting) `index.html` als den Namen des Index document (Indexdokuments) eingeben, muss der Dateiname des Indexdokuments ebenfalls `index.html` und nicht `Index.html` lauten.

3. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Indexdokuments ein (z. B. `index.html`). Weitere Informationen finden Sie unter [Aktivieren des Website-Hostings](#).

Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Indexdokument in Ihren Bucket hochzuladen:
 - Ziehen Sie die Indexdatei per Drag & Drop in das Konsolen-Bucket-Verzeichnis.
 - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-step Anweisungen finden Sie unter [Objekte hochladen](#).

7. (Optional) Laden Sie andere Website-Inhalte in Ihren Bucket hoch.

Schritt 6: Konfigurieren eines Fehlerdokuments

Wenn Sie das Hosting statischer Websites für Ihren Bucket aktivieren, geben Sie den Namen des Fehlerdokuments ein (z. B. **404.html**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Fehlerdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie ein Fehlerdokument

1. Erstellen Sie ein Fehlerdokument, z. B. `404.html`.
2. Speichern Sie die Fehlerdokumentdatei lokal.

Der Name des Fehlerdokuments unterscheidet zwischen Groß- und Kleinschreibung und muss genau mit dem Namen übereinstimmen, den Sie beim Aktivieren des statischen Website-

Hostings eingeben. Wenn Sie beispielsweise `404.html` im Dialogfeld **Hosten** einer statischen Website als Namen des Fehlerdokuments eingeben, muss der Dateiname des Fehlerdokuments ebenfalls `404.html` lauten.

3. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Fehlerdokuments ein (z. B. `404.html`). Weitere Informationen erhalten Sie unter [Aktivieren des Website-Hostings](#) und [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#).

Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Fehlerdokument in Ihren Bucket hochzuladen:
 - Ziehen Sie die Fehlerdokumentdatei in das Konsolen-Bucket-Verzeichnis.
 - Wählen Sie **Upload (Hochladen)** und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-step Anweisungen finden Sie unter [Objekte hochladen](#).

Schritt 7: Testen des Website-Endpunkts

Nach der Konfigurierung des statischen Website-Hostings für Ihren Bucket können Sie den Website-Endpunkt testen.

Note

Amazon S3 unterstützt keinen HTTPS Zugriff auf die Website. Wenn Sie verwenden möchten HTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird.

Weitere Informationen finden Sie unter [Wie verwende ich, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird?](#) und [Erforderlich HTTPS für die Kommunikation zwischen Zuschauern und CloudFront](#).

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Properties (Eigenschaften).
3. Wählen Sie unten auf der Seite unter Static website hosting (Hosting statischer Websites) Ihren Bucket-Website-Endpunkt.

Ihr Indextdokument wird in einem separaten Browserfenster geöffnet.

Sie haben jetzt eine auf Amazon S3 gehostete Website. Diese Website steht am Amazon S3-Website-Endpunkt zur Verfügung. Möglicherweise haben Sie jedoch eine Domäne wie `example.com`, die den Inhalt von der von Ihnen erstellten Website bereitstellen soll. Sie könnten auch die Amazon S3-Unterstützung der Root-Domäne nutzen, um Anforderungen für `http://www.example.com` und `http://example.com` zu bedienen. Dafür sind zusätzliche Schritte erforderlich. Ein Beispiel finden Sie unter [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#).

Schritt 8: Bereinigen

Wenn Sie Ihre statische Website nur zu Lernzwecken erstellt haben, löschen Sie die AWS Ressourcen, die Sie so zugewiesen haben, dass Ihnen keine Gebühren mehr anfallen. Nachdem Sie Ihre gelöscht haben AWS Ressourcen, Ihre Website ist nicht mehr verfügbar. Weitere Informationen finden Sie unter [Löschen eines Buckets](#).

Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne

Angenommen, Sie wollen eine statische Website auf Amazon S3 hosten. Sie haben eine Domäne bei Amazon Route 53 registriert (zum Beispiel `example.com`), und Sie möchten, dass Anforderungen für `http://www.example.com` und `http://example.com` von Ihren Amazon-S3-Inhalten aus bedient werden. Sie können diese Anleitung verwenden, um zu erfahren, wie Sie eine statische Website hosten und Umleitungen auf Amazon S3 für eine Website mit einem benutzerdefinierten Domänennamen, der bei Route 53 registriert ist, anlegen können. Sie können mit einer vorhandenen Website arbeiten, die Sie auf Amazon S3 hosten möchten, oder diesen Walkthrough verwenden, um bei Null anzufangen.

Nachdem Sie diese exemplarische Vorgehensweise abgeschlossen haben, können Sie optional Amazon verwenden CloudFront , um die Leistung Ihrer Website zu verbessern. Weitere Informationen finden Sie unter [Beschleunigen Sie Ihre Website mit Amazon CloudFront](#).

Note

Endpunkte der Amazon S3 S3-Website unterstützen keine HTTPS Zugriffspunkte. Wenn Sie verwenden möchten HTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird.

Ein Tutorial zum sicheren Hosten Ihrer Inhalte mit CloudFront Amazon S3 finden Sie unter [Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53](#). Weitere Informationen finden Sie unter [Wie verwende ich, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird?](#) und [Erforderlich HTTPS für die Kommunikation zwischen Zuschauern und CloudFront](#).

Automatisierung der statischen Website-Einrichtung mit einem AWS CloudFormation Vorlage

Sie können eine verwenden AWS CloudFormation Vorlage zur Automatisierung der Einrichtung Ihrer statischen Website. Das Tool AWS CloudFormation Die Vorlage richtet die Komponenten ein, die Sie zum Hosten einer sicheren statischen Website benötigen, sodass Sie sich mehr auf den Inhalt Ihrer Website und weniger auf die Konfiguration von Komponenten konzentrieren können.

Das Tool AWS CloudFormation Die Vorlage umfasst die folgenden Komponenten:

- Amazon S3 – Erstellt einen Amazon-S3-Bucket zum Hosten Ihrer statischen Website.
- CloudFront — Erstellt eine CloudFront Distribution, um Ihre statische Website zu beschleunigen.
- Lambda@Edge – Verwendet [Lambda@Edge](#), um jeder Serverantwort Sicherheits-Header hinzuzufügen. Sicherheits-Header sind eine Gruppe von Headern in der Webserverantwort, die Webbrowser anweisen, zusätzliche Sicherheitsvorkehrungen zu treffen. Weitere Informationen finden Sie im Blogbeitrag [Hinzufügen von HTTP Sicherheitsheadern mithilfe von Lambda @Edge und Amazon](#). CloudFront

Dieser AWS CloudFormation Die Vorlage steht Ihnen zum Herunterladen und Verwenden zur Verfügung. Informationen und Anweisungen finden Sie unter [Erste Schritte mit einer sicheren statischen Website](#) im Amazon CloudFront Developer Guide.

Themen

- [Bevor Sie beginnen](#)
- [Schritt 1: Registrieren einer benutzerdefinierten Domäne bei Route 53](#)
- [Schritt 2: Erstellen von zwei Buckets](#)

- [Schritt 3: Konfigurieren Ihres Stammdomänen-Buckets für Website-Hosting](#)
- [Schritt 4: Konfigurieren Ihres Subdomänen-Buckets für die Website-Umleitung](#)
- [Schritt 5: Konfigurieren der Protokollierung für Website-Datenverkehr](#)
- [Schritt 6: Hochladen des Index und des Website-Inhalts](#)
- [Schritt 7: Hochladen eines Fehlerdokuments](#)
- [Schritt 8: Bearbeiten der S3 Block Public Access-Einstellungen](#)
- [Schritt 9: Anfügen einer Bucket-Richtlinie](#)
- [Schritt 10: Testen Ihres Domänen-Endpunkts](#)
- [Schritt 11: Hinzufügen von Aliasdatensätzen für Ihre Domäne und Subdomäne](#)
- [Schritt 12: Testen der Website](#)
- [Beschleunigen Sie Ihre Website mit Amazon CloudFront](#)
- [Bereinigung Ihrer Beispielressourcen](#)

Bevor Sie beginnen

Während Sie den Schritten in diesem Beispiel folgen, arbeiten Sie mit den folgenden Services:

Amazon Route 53 – Sie verwenden Route 53, um Domänen zu registrieren und zu definieren, wohin der Internetdatenverkehr für Ihre Domäne geleitet werden soll. Das Beispiel zeigt, wie Sie Route 53-Aliaseinträge erstellen, die den Verkehr für Ihre Domain (`example.com`) und Subdomain (`www.example.com`) an einen Amazon S3-Bucket weiterleiten, der eine HTML Datei enthält.

Amazon S3 – Mit Amazon S3 erstellen Sie Buckets, laden eine Beispielseite für eine Website hoch, konfigurieren Berechtigungen, sodass jeder den Inhalt sehen kann, und konfigurieren dann die Buckets für das Website-Hosting.

Schritt 1: Registrieren einer benutzerdefinierten Domäne bei Route 53

Wenn Sie noch keinen registrierten Domänennamen haben, wie z. B. `example.com`, registrieren Sie einen mit Route 53. Weitere Informationen finden Sie unter [Registrieren einer neuen Domäne](#) im Amazon-Route 53-Entwicklerhandbuch. Nachdem Sie Ihren Domänennamen registriert haben, können Sie Ihre Amazon-S3-Buckets für das Website-Hosting erstellen und konfigurieren.

Schritt 2: Erstellen von zwei Buckets

Um Anforderungen von sowohl der Stammdomäne als auch von der Subdomäne zu unterstützen, erstellen Sie zwei Buckets.

- Domänen-Bucket – `example.com`
- Subdomänen-Bucket – `www.example.com`

Diese Bucket-Namen müssen genau mit Ihrem Domänennamen übereinstimmen. In diesem Beispiel lautet der Domänenname `example.com`. Sie hosten Ihre Inhalte aus dem Stammdomänen-Bucket (`example.com`). Sie erstellen eine Umleitungsanforderung für den Subdomänen-Bucket (`www.example.com`). Falls ein Benutzer `www.example.com` in seinen Browser eingibt, wird er zu `example.com` umgeleitet und sieht den Inhalt, der im Amazon-S3-Bucket mit diesem Namen gehostet wird.

So erstellen Sie Ihre Buckets für Website-Hosting

Die folgenden Anweisungen geben einen Überblick darüber, wie Sie Ihre Buckets für das Website-Hosting erstellen. Eine ausführliche step-by-step Anleitung zur Erstellung eines Buckets finden Sie unter [Erstellen eines Buckets](#).

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Erstellen Sie den Stammdomänen-Bucket:
 - a. Wählen Sie Create Bucket (Bucket erstellen) aus.
 - b. Geben Sie den Bucket-Namen ein (z. B. **example.com**).
 - c. Wählen Sie die Region aus, in der Sie Ihren Bucket erstellen möchten.

Wählen Sie eine Region in Ihrer Nähe aus, um Latenz und Kosten auf einem Minimum zu halten oder behördliche Vorschriften zu erfüllen. Die von Ihnen ausgewählte Region bestimmt Ihren Amazon-S3-Website-Endpunkt. Weitere Informationen finden Sie unter [Website-Endpunkte](#).

- d. Um die Standardeinstellungen zu übernehmen und den Bucket zu erstellen, wählen Sie Create (Erstellen).
3. Erstellen Sie Ihren Subdomänen-Bucket:

- a. Wählen Sie Create Bucket (Bucket erstellen) aus.
- b. Geben Sie den Bucket-Namen ein (z. B. **www.example.com**).
- c. Wählen Sie die Region aus, in der Sie Ihren Bucket erstellen möchten.

Wählen Sie eine Region in Ihrer Nähe aus, um Latenz und Kosten auf einem Minimum zu halten oder behördliche Vorschriften zu erfüllen. Die von Ihnen ausgewählte Region bestimmt Ihren Amazon-S3-Website-Endpunkt. Weitere Informationen finden Sie unter [Website-Endpunkte](#).

- d. Um die Standardeinstellungen zu übernehmen und den Bucket zu erstellen, wählen Sie Create (Erstellen).

Im nächsten Schritt konfigurieren Sie `example.com` für das Website-Hosting.

Schritt 3: Konfigurieren Ihres Stammdomänen-Buckets für Website-Hosting

In diesem Schritt konfigurieren Sie Ihren Stammdomänen-Bucket (`example.com`) als Website. Dieser Bucket enthält Ihren Website-Inhalt. Wenn Sie einen Bucket für das Website-Hosting konfigurieren, können Sie über auf die Website zugreife [Website-Endpunkte](#).

So aktivieren Sie das statische Website-Hosting

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie das Hosting statischer Websites aktivieren wollen.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Wählen Sie Use this bucket to host a website (Diesen Bucket zum Hosten einer Website verwenden).
6. Wählen Sie unter Static website hosting (Hosting statischer Websites) die Option Enable (Aktivieren) aus
7. Geben Sie unter Index document (Index-Dokument) den Dateinamen des Index-Dokuments ein, der typischerweise `index.html` ist.

Beim Namen des Indextdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss exakt mit dem Dateinamen des HTML Indextdokuments übereinstimmen, das Sie

in Ihren S3-Bucket hochladen möchten. Wenn Sie Ihren Bucket für das Hosting von Websites konfigurieren, müssen Sie ein Indextdokument angeben. Amazon S3 gibt dieses Indextdokument zurück, wenn Anfragen an die Root-Domäne oder einen der Unterordner gestellt werden. Weitere Informationen finden Sie unter [Konfigurieren eines Indextdokuments](#).

8. Um ein eigenes benutzerdefiniertes Fehlerdokument für Fehler der Klasse 4XX bereitzustellen, geben Sie unter Fehlerdokument den Dateinamen des benutzerdefinierten Fehlerdokuments ein.

Beim Namen des Fehlerdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss genau mit dem Dateinamen des HTML Fehlerdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie kein benutzerdefiniertes Fehlerdokument angeben und ein Fehler auftritt, gibt Amazon S3 ein HTML Standard-Fehlerdokument zurück. Weitere Informationen finden Sie unter [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#).

9. (Optional) Wenn Sie erweiterte Umleitungsregeln angeben möchten, geben Sie im Feld JSONUmleitungsregeln eine Beschreibung der Regeln ein.

Beispielsweise können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten. Weitere Informationen finden Sie unter [Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen](#).

10. Wählen Sie Save Changes (Änderungen speichern).

Amazon S3 ermöglicht statisches Website-Hosting für Ihren Bucket. Unten auf der Seite sehen Sie unter Static website hosting (Hosting statischer Websites) den Website-Endpoint für Ihren Bucket.

11. Notieren Sie unter Static website hosting (Statisches Website-Hosting) den Wert für Endpoint (Endpoint).

Der Endpoint (Endpoint) ist der Amazon-S3-Website-Endpoint für Ihren Bucket. Nachdem Sie den Bucket als statische Website konfiguriert haben, können Sie diesen Endpoint verwenden, um Ihre Website zu testen.

Nachdem Sie die [Blockierungseinstellungen für den öffentlichen Zugriff bearbeitet](#) und [eine Bucket-Richtlinie hinzugefügt](#) haben, die öffentlichen Lesezugriff ermöglicht, können Sie den Website-Endpoint verwenden, um auf Ihre Website zuzugreifen.

Im nächsten Schritt konfigurieren Sie Ihre Subdomäne (`www.example.com`) zur Weiterleitung von Anforderungen an Ihre Domäne (`example.com`).

Schritt 4: Konfigurieren Ihres Subdomänen-Buckets für die Website-Umleitung

Nachdem Sie Ihren Stammdomänen-Bucket für das Website-Hosting konfiguriert haben, können Sie Ihren Subdomänen-Bucket so konfigurieren, dass alle Anforderungen zu der Domäne umgeleitet werden. In diesem Beispiel werden alle Anforderungen für `www.example.com` an `example.com` umgeleitet.

So konfigurieren Sie eine Umleitungsanforderung

1. Wählen Sie in der Amazon-S3-Konsole in der Liste Buckets Ihren Subdomänen-Bucket aus (in diesem Beispiel `www.example.com`).
2. Wählen Sie Properties (Eigenschaften).
3. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
4. Wählen Sie Redirect requests for an object (Anfragen für ein Objekt umleiten).
5. Geben Sie im Feld Target bucket (Ziel-Bucket) Ihre Root-Domäne ein, z. B. **example.com**.
6. Wählen Sie für Protocol (Protokoll) die Option http aus.
7. Wählen Sie Save Changes (Änderungen speichern).

Schritt 5: Konfigurieren der Protokollierung für Website-Datenverkehr

Wenn Sie die Anzahl der Besucher nachverfolgen möchten, die auf Ihre Website zugreifen, können Sie optional die Protokollierung für Ihren Stammdomänen-Bucket aktivieren. Weitere Informationen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#). Wenn Sie Amazon verwenden möchten, CloudFront um Ihre Website zu beschleunigen, können Sie auch die CloudFront Protokollierung verwenden.

So aktivieren Sie die Server-Zugriffsprotokollierung für Ihren Stammdomänen-Bucket

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Erstellen Sie in derselben Region, in der Sie den Bucket erstellt haben, der als statische Website konfiguriert ist, einen Bucket für die Protokollierung, z. B. `logs.example.com`.
3. Erstellen Sie einen Ordner für die Protokolldateien der Server-Zugriffsprotokollierung (z. B. `logs`).
4. (Optional) Wenn Sie die Leistung Ihrer Website verbessern CloudFront möchten, erstellen Sie einen Ordner für die CloudFront Protokolldateien (z. B. `cdn`).

⚠ Important

Wenn Sie eine Distribution erstellen oder aktualisieren und die CloudFront Protokollierung aktivieren, CloudFront aktualisiert die Bucket-Zugriffskontrollliste (ACL), sodass das `awslogsdelivery` Konto `FULL_CONTROL` berechtigt ist, Logs in Ihren Bucket zu schreiben. Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Konfiguration der Standardprotokollierung und für den Zugriff auf Ihre Protokolldateien](#) im Amazon CloudFront Developer Guide. Wenn der Bucket, der die Logs speichert, die vom Bucket-Besitzer erzwungene Einstellung zur Deaktivierung von S3 Object Ownership verwendet, CloudFront können keine Logs in den Bucket geschrieben werden. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

5. Wählen Sie in der Liste Buckets Ihren Stammdomänen-Bucket aus.
6. Wählen Sie Properties (Eigenschaften).
7. Wählen Sie unter Server access logging (Server-Zugriffsprotokollierung) Edit (Bearbeiten).
8. Wählen Sie Enable aus.
9. Wählen Sie im Ziel-Bucket den Bucket und das Ordnerziel für die Server-Zugriffsprotokolle aus:
 - Navigieren Sie zum Ordner- und Bucket-Speicherort:
 1. Wählen Sie Browse S3 (S3 durchsuchen).
 2. Wählen Sie den Bucket-Namen und dann den Ordner mit den Protokollen aus.
 3. Wählen Sie Choose path (Pfad wählen).
 - Geben Sie den S3-Bucket-Pfad ein, z. B. `s3://logs.example.com/logs/`.
10. Wählen Sie Save Changes (Änderungen speichern).

In Ihrem Protokoll-Bucket können Sie jetzt auf Ihre Protokolle zugreifen. Amazon S3 schreibt Website-Zugriffsprotokolle alle zwei Stunden in Ihren Bucket zur Protokollierung.

Schritt 6: Hochladen des Index und des Website-Inhalts

Als Nächstes laden Sie das Indextokument und den optionalen Website-Inhalt zum Stammdomänen-Bucket hoch.

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indextdokuments ein (z. B, **index.html**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Indextdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie das Indextdokument

1. Erstellen Sie eine Datei `index.html`.

Wenn Sie noch keine `index.html` Datei haben, können Sie wie folgt HTML eine Datei erstellen:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>My Website Home Page</title>
</head>
<body>
  <h1>Welcome to my website</h1>
  <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

2. Speichern Sie die Indexdatei lokal.

Der Dateiname des Indextdokuments muss genau mit dem Namen des Indextdokuments übereinstimmen, den Sie im Dialogfeld Static website hosting (Statisches Website-Hosting) eingeben. Beim Namen des Indextdokuments wird die Groß- und Kleinschreibung berücksichtigt. Wenn Sie beispielsweise im Dialogfeld Static website hosting (Statisches Website-Hosting) `index.html` als den Namen des Index document (Indextdokuments) eingeben, muss der Dateiname des Indextdokuments ebenfalls `index.html` und nicht `Index.html` lauten.

3. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Indextdokuments ein (z. B. `index.html`). Weitere Informationen finden Sie unter [Aktivieren des Website-Hostings](#).

Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Indexdokument in Ihren Bucket hochzuladen:
 - Ziehen Sie die Indexdatei per Drag & Drop in das Konsolen-Bucket-Verzeichnis.
 - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-step Anweisungen finden Sie unter [Objekte hochladen](#).

7. (Optional) Laden Sie andere Website-Inhalte in Ihren Bucket hoch.

Schritt 7: Hochladen eines Fehlerdokuments

Wenn Sie das Hosting statischer Websites für Ihren Bucket aktivieren, geben Sie den Namen des Fehlerdokuments ein (z. B, **404.html**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Fehlerdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie ein Fehlerdokument

1. Erstellen Sie ein Fehlerdokument, z. B. `404.html`.
2. Speichern Sie die Fehlerdokumentdatei lokal.

Der Name des Fehlerdokuments unterscheidet zwischen Groß- und Kleinschreibung und muss genau mit dem Namen übereinstimmen, den Sie beim Aktivieren des statischen Website-Hostings eingeben. Wenn Sie beispielsweise `404.html` im Dialogfeld Hosten einer statischen Website als Namen des Fehlerdokuments eingeben, muss der Dateiname des Fehlerdokuments ebenfalls `404.html` lauten.

3. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Fehlerdokuments ein (z. B, `404.html`). Weitere Informationen erhalten Sie unter [Aktivieren des Website-Hostings](#) und [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#).

Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Fehlerdokument in Ihren Bucket hochzuladen:
 - Ziehen Sie die Fehlerdokumentdatei in das Konsolen-Bucket-Verzeichnis.
 - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-step Anweisungen finden Sie unter [Objekte hochladen](#).

Schritt 8: Bearbeiten der S3 Block Public Access-Einstellungen

In diesem Beispiel bearbeiten Sie die Block-Einstellungen für den öffentlichen Zugriff für den Domänen-Bucket (example.com), um den öffentlichen Zugriff zu ermöglichen.


Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

Warning

Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#) dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugangs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#) 



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

- Block *all* public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

 - Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
 - Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.
 - Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
 - Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die [Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten](#), bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Schritt 9: Anfügen einer Bucket-Richtlinie

In diesem Beispiel hängen Sie eine Bucket-Richtlinie an den Domain-Bucket (example.com) an, um öffentlichen Lesezugriff zu ermöglichen. Sie ersetzen den *Bucket-Name* in der Beispiel-Bucket-Richtlinie zum Beispiel mit dem Namen Ihres Domain-Bucketsexample.com.

Nachdem Sie die Einstellungen für S3 Block Public Access bearbeitet haben, können Sie eine Bucket-Richtlinie hinzufügen, um öffentlichen Lesezugriff auf den Bucket zu gewähren. Wenn Sie öffentlichen Lesezugriff gewähren, kann jeder im Internet auf Ihren Bucket zugreifen.

Important

Die zuvor genannte Richtlinie ist nur ein Beispiel und erlaubt Vollzugriff auf die Inhalte Ihres Buckets. Bevor Sie mit diesem Schritt fortfahren, lesen Sie den Abschnitt [Wie kann ich die Dateien in meinem Amazon-S3-Bucket sichern?](#), um sicherzustellen, dass Sie die bewährten Methoden zum Sichern der Dateien in Ihrem S3-Bucket und die Risiken in Zusammenhang mit der Gewährung von öffentlichem Zugriff kennen.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Bucket Policy (Bucket-Richtlinie) Edit (Bearbeiten).
4. Um öffentlichen Lesezugriff auf Ihre Website zu gewähren, kopieren Sie die folgende Bucket-Richtlinie und fügen Sie sie in den Bucket policy editor (Bucket-Richtlinieneditor) ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::Bucket-Name/*"
      ]
    }
  ]
}
```

5. Aktualisieren Sie den Resource zu Ihrem Bucket-Namen.

Im vorherigen Beispiel für eine Bucket-Richtlinie *Bucket-Name* ist ein Platzhalter für den Bucket-Namen. Um diese Bucket-Richtlinie mit Ihrem eigenen Bucket zu verwenden, müssen Sie diesen Namen so aktualisieren, dass er mit Ihrem Bucket übereinstimmt.

6. Wählen Sie Save Changes (Änderungen speichern).

Es wird eine Meldung angezeigt, die darauf hinweist, dass die Bucket-Richtlinie erfolgreich hinzugefügt wurde.

Wenn die Fehlermeldung `Policy has invalid resource` angezeigt wird, bestätigen Sie, dass der Bucket-Name in der Bucket-Richtlinie mit Ihrem Bucket-Namen übereinstimmt. Informationen zum Hinzufügen einer Bucket-Richtlinie finden Sie unter [Wie füge ich eine S3-Bucket-Richtlinie hinzu?](#)

Wenn Sie eine Fehlermeldung erhalten und die Bucket-Richtlinie nicht speichern können, überprüfen Sie Ihr Konto und die Bucket-Einstellungen für Block Public Access, um zu bestätigen, dass Sie den öffentlichen Zugriff auf den Bucket zulassen.

Im nächsten Schritt können Sie die Endpunkte Ihrer Website herausfinden und Ihren Domänen-Endpunkt testen.

Schritt 10: Testen Ihres Domänen-Endpunkts

Nachdem Sie den Stammdomänen-Bucket zum Hosten einer öffentlichen Website konfiguriert haben, können Sie Ihren Endpunkt testen. Weitere Informationen finden Sie unter [Website-Endpunkte](#). Sie können nur den Endpunkt für Ihren Domänen-Bucket testen, da Ihr Subdomänen-Bucket für die Website-Umleitung und nicht für das statische Website-Hosting eingerichtet ist.

Note

Amazon S3 unterstützt keinen HTTPS Zugriff auf die Website. Wenn Sie verwenden möchten HTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird.

Weitere Informationen finden Sie unter [Wie verwende ich, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird?](#) und [Erforderlich HTTPS für die Kommunikation zwischen Zuschauern und CloudFront](#).

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Properties (Eigenschaften).
3. Wählen Sie unten auf der Seite unter Static website hosting (Hosting statischer Websites) Ihren Bucket-Website-Endpunkt.

Ihr Indextokument wird in einem separaten Browserfenster geöffnet.

Im nächsten Schritt verwenden Sie Amazon Route 53, damit Kunden Ihre beiden benutzerdefinierten Optionen verwenden können URLs, um zu Ihrer Website zu navigieren.

Schritt 11: Hinzufügen von Aliasdatensätzen für Ihre Domäne und Subdomäne

In diesem Schritt erstellen Sie die Aliasdatensätze, die Sie der gehosteten Zone für Ihre Domänenzuordnungen `example.com` und `www.example.com` hinzufügen. Statt IP-Adressen verwenden die Alias-Datensätze Amazon S3 Website-Endpunkte. Amazon Route 53 verwaltet ein Mapping zwischen den Alias-Datensätzen und den IP-Adressen, wo sich die Amazon-S3-Buckets befinden. Sie erstellen zwei Alias-Datensätze: einen für Ihre Stamm-Domäne und einen für Ihre Subdomäne.

Hinzufügen eines Aliasdatensatzes für Ihre Stamm-Domäne und Subdomäne


So fügen Sie Ihrer Stamm-Domäne einen Aliasdatensatz hinzu (**example.com**)

1. Öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.

Note

Wenn Sie Route 53 noch nicht verwenden, lesen Sie [Schritt 1: Registrieren einer Domäne](#) im Amazon-Route 53-Entwicklerhandbuch. Nachdem Sie Ihre Einrichtung abgeschlossen haben, können Sie mit den Anweisungen fortfahren.

2. Wählen Sie Hosted Zones (Gehostete Zonen) aus.
3. Wählen Sie in der Liste der gehosteten Zonen den Namen der gehosteten Zone aus, der dem Domänennamen entspricht.
4. Wählen Sie Create Record Set (Datensatz erstellen).
5. Wählen Sie Switch to wizard (Zu Assistent wechseln) aus.

 Note

Wenn Sie Quick Create zum Erstellen Ihrer Aliaseinträge verwenden möchten, lesen Sie [Konfigurieren von Route 53 zum Weiterleiten von Datenverkehr an einen S3-Bucket](#).

6. Wählen Sie Simple Routing (Einfaches Routing), und wählen Sie Next (Weiter).
7. Wählen Sie Define simple record (Einfachen Datensatz definieren).
8. Akzeptieren Sie unter Record name (Datensatzname) den Standardwert, bei dem es sich um den Namen Ihrer gehosteten Zone und Ihrer Domäne handelt.
9. Wählen Sie unter Value/Route traffic to (Wert/Datenverkehr weiterleiten zu) die Option Alias to S3 website endpoint (Alias zu S3-Website-Endpunkt) aus.
10. Wählen Sie die Region aus.
11. Wählen Sie den S3-Bucket.

Der Bucket-Name sollte mit dem Namen übereinstimmen, der im Feld Name angezeigt wird. In der Liste Choose S3 Bucket (S3-Bucket auswählen) wird der Bucket-Name mit dem Amazon-S3-Website-Endpunkt für die Region angezeigt, in der der Bucket erstellt wurde, zum Beispiel `s3-website-us-west-1.amazonaws.com` (`example.com`).

Choose S3 bucket (S3-Bucket auswählen) listet einen Bucket auf, wenn:

- Sie den Bucket als statische Website konfiguriert haben.
- Der Name des Buckets mit dem Namen des Datensatzes übereinstimmt, den Sie anlegen.
- Das aktuelle AWS-Konto hat den Bucket erstellt.

Wenn Ihr Bucket nicht in der Auflistung Choose S3 bucket (S3-Bucket auswählen) angezeigt wird, geben Sie den Amazon-S3-Website-Endpunkt für die Region ein, in der der Bucket erstellt wurde, z. B. **`s3-website-us-west-2.amazonaws.com`**. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter [Amazon-S3-Website-Endpunkte](#). Weitere Informationen über das Alias-Target finden Sie unter [Wert/Datenverkehr weiterleiten zu](#) im Amazon-Route 53-Entwicklerhandbuch.

12. Wählen Sie unter Datensatztyp die Option A - Leitet den Verkehr an eine IPv4 Adresse und einige AWS Ressourcen.
13. Wählen Sie unter Evaluate target health (Zielzustand bewerten) die Option No (Nein).
14. Wählen Sie Define simple record (Einfachen Datensatz definieren).

So fügen Sie Ihrer Subdomäne () einen Alias-Datensatz hi (**www.example.com**)

1. Wählen Sie unter Configure records (Datensätze konfigurieren) die Option Define simple record (Einfachen Datensatz definieren) aus
2. Geben Sie unter Record name (Datensatzname) für Ihre Subdomäne `www` ein.
3. Wählen Sie unter Value/Route traffic to (Wert/Datenverkehr weiterleiten zu) die Option Alias to S3 website endpoint (Alias zu S3-Website-Endpoint) aus.
4. Wählen Sie die Region aus.
5. Wählen Sie den S3-Bucket, zum Beispiel, `s3-website-us-west-2.amazonaws.com` (`www.example.com`).

Wenn Ihr Bucket nicht in der Auflistung Choose S3 bucket (S3-Bucket auswählen) angezeigt wird, geben Sie den Amazon-S3-Website-Endpoint für die Region ein, in der der Bucket erstellt wurde, z. B. **s3-website-us-west-2.amazonaws.com**. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter [Amazon-S3-Website-Endpunkte](#). Weitere Informationen über das Alias-Target finden Sie unter [Wert/Datenverkehr weiterleiten zu](#) im Amazon-Route 53-Entwicklerhandbuch.

6. Wählen Sie unter Datensatztyp die Option A - Leitet den Verkehr an eine IPv4 Adresse und einige AWS Ressourcen.
7. Wählen Sie unter Evaluate target health (Zielzustand bewerten) die Option No (Nein).
8. Wählen Sie Define simple record (Einfachen Datensatz definieren).
9. Klicken Sie auf der Seite Configure records (Datensätze konfigurieren) auf Create records (Datensätze erstellen).

Note

Änderungen werden im Allgemeinen innerhalb von 60 Sekunden an alle Route 53-Server übertragen. Wenn die Übertragung abgeschlossen ist, können Sie den Datenverkehr an den Amazon-S3-Bucket leiten, indem Sie den Namen des Alias-Datensatzes angeben, den Sie in diesem Verfahren erstellt haben.

Hinzufügen eines Alias-Datensatzes für Ihre Stamm-Domäne und Subdomäne (alte Route 53-Konsole)

So fügen Sie Ihrer Stamm-Domäne einen Aliasdatensatz hinzu (**example.com**)

Die Route 53-Konsole wurde überarbeitet. In der Route 53-Konsole können Sie vorübergehend die alte Konsole verwenden. Wenn Sie sich für die Arbeit mit der alten Route 53-Konsole entscheiden, gehen Sie wie folgt vor.

1. Öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.

Note

Wenn Sie Route 53 noch nicht verwenden, lesen Sie [Schritt 1: Registrieren einer Domäne](#) im Amazon-Route 53-Entwicklerhandbuch. Nachdem Sie Ihre Einrichtung abgeschlossen haben, können Sie mit den Anweisungen fortfahren.

2. Wählen Sie Hosted Zones (Gehostete Zonen) aus.
3. Wählen Sie in der Liste der gehosteten Zonen den Namen der gehosteten Zone aus, der dem Domänennamen entspricht.
4. Wählen Sie Create Record Set (Datensatz erstellen).
5. Geben Sie die folgenden Werte an:

Name

Akzeptieren Sie den Standardwert, bei dem es sich um den Namen Ihrer gehosteten Zone und Ihrer Domäne handelt.

Für die Stamm-Domäne müssen Sie keine zusätzlichen Informationen in das Feld Name eingeben.

Typ

Wählen Sie A — IPv4 Adresse.

Alias

Wählen Sie Yes (Ja).

Alias-Ziel

Wählen Sie im Bereich S3 website endpoints (S3-Website-Endpunkte) der Liste Ihren Bucket-Namen aus.

Der Bucket-Name sollte mit dem Namen übereinstimmen, der im Feld Name angezeigt wird. In der Auflistung Alias Target (Alias-Ziel) folgt dem Bucket-Namen beispielsweise der Amazon-S3-Website-Endpunkt für die Region, in der der Bucket erstellt wurde, z. B. `example.com (s3-website-us-west-2.amazonaws.com)`. Alias Target (Alias-Ziel) listet einen Bucket auf, wenn:

- Sie den Bucket als statische Website konfiguriert haben.
- Der Name des Buckets mit dem Namen des Datensatzes übereinstimmt, den Sie anlegen.
- Die aktuelle AWS-Konto hat den Bucket erstellt.

Wenn Ihr Bucket nicht in der Auflistung Alias Target (Alias-Ziel) angezeigt wird, geben Sie den Amazon-S3-Website-Endpunkt für die Region ein, in der der Bucket erstellt wurde, z. B. `s3-website-us-west-2`. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter [Amazon-S3-Website-Endpunkte](#). Weitere Informationen über das Alias-Target finden Sie unter [Value/Route Traffic to \(Wert/Datenverkehr weiterleiten zu\)](#) im Amazon-Route 53-Entwicklerhandbuch.

Routing-Richtlinie

Übernehmen Sie den Standardwert Simple.

Evaluate Target Health

Übernehmen Sie den Standardwert No.

6. Wählen Sie Create (Erstellen) aus.

So fügen Sie Ihrer Subdomäne () einen Alias-Datensatz hi (**www.example.com**)

1. Wählen Sie in der gehosteten Zone für Ihre Stamm-Domäne (`example.com`) die Option Create record set (Datensatz erstellen) aus.
2. Geben Sie die folgenden Werte an:

Name

Geben Sie als Subdomäne „www“ in das Feld ein.

Typ

Wählen Sie A — IPv4 Adresse.

Alias

Wählen Sie Yes (Ja).

Alias-Ziel

Wählen Sie im Abschnitt S3 website endpoints (S3-Website-Endpunkte) der Liste denselben Bucket-Namen aus, der im Feld Name angezeigt wird (z. B. `www.example.com` (`s3-website-us-west-2.amazonaws.com`)).

Routing-Richtlinie

Übernehmen Sie den Standardwert Simple.

Evaluate Target Health

Übernehmen Sie den Standardwert No.

3. Wählen Sie Create aus.

Note

Änderungen werden im Allgemeinen innerhalb von 60 Sekunden an alle Route 53-Server übertragen. Wenn die Übertragung abgeschlossen ist, können Sie den Datenverkehr an den Amazon-S3-Bucket leiten, indem Sie den Namen des Alias-Datensatzes angeben, den Sie in diesem Verfahren erstellt haben.

Schritt 12: Testen der Website

Bestätigen Sie, dass die Website und die Umleitung korrekt funktionieren. Geben Sie in Ihrem Browser Ihre einURLs. In diesem Beispiel können Sie Folgendes versuchenURLs:

- Domain (Domäne) (`http://example.com`) – Zeigt das Indextdokument im `example.com`-Bucket an.
- Subdomain (Subdomäne) (`http://www.example.com`) – Leitet Ihre Anforderung an `http://example.com` weiter. Sie sehen das Indextdokument im `example.com`-Bucket.

Wenn Ihre Website oder Weiterleitungslinks nicht funktionieren, können Sie Folgendes versuchen:

- Clear cache (Cache löschen) – Löschen Sie den Cache Ihres Webbrowsers.
- Check name servers (Nameserver überprüfen) – Wenn Ihre Webseite und Weiterleitungslinks nach dem Löschen des Caches nicht funktionieren, können Sie die Namenserver für Ihre Domäne und die Namenserver für Ihre gehostete Zone vergleichen. Wenn die Namenserver nicht übereinstimmen, müssen Sie möglicherweise Ihre Domänennamenserver so aktualisieren, dass sie mit den unter Ihrer gehosteten Zone aufgelisteten übereinstimmen. Weitere Informationen finden Sie unter [Hinzufügen oder Ändern von Namenservern und Kleben von Datensätzen für eine Domäne](#).

Nachdem Sie Ihre Stammdomain und Subdomain erfolgreich getestet haben, können Sie eine [CloudFrontAmazon-Distribution](#) einrichten, um die Leistung Ihrer Website zu verbessern und Protokolle bereitzustellen, anhand derer Sie den Website-Traffic überprüfen können. Weitere Informationen finden Sie unter [Beschleunigen Sie Ihre Website mit Amazon CloudFront](#).

Beschleunigen Sie Ihre Website mit Amazon CloudFront

Sie können [Amazon](#) verwenden CloudFront, um die Leistung Ihrer Amazon S3 S3-Website zu verbessern. CloudFront stellt Ihre Website-Dateien (wie HTML Bilder und Videos) in Rechenzentren auf der ganzen Welt (sogenannte Edge-Standorte) zur Verfügung. Wenn ein Besucher eine Datei von Ihrer Website anfordert, leitet er die Anfrage CloudFront automatisch an eine Kopie der Datei am nächstgelegenen Edge-Standort weiter. Das führt zu schnelleren Download-Zeiten, als wenn der Besucher den Inhalt von einem weitere entfernten Rechenzentrum angefordert hätte.

CloudFront speichert Inhalte an Edge-Standorten für einen von Ihnen angegebenen Zeitraum im Cache. Wenn ein Besucher Inhalte anfordert, die länger als das Ablaufdatum zwischengespeichert wurden, überprüft er auf dem Ursprungsserver, ob eine neuere Version des Inhalts verfügbar ist. Wenn eine neuere Version verfügbar ist, wird die neue Version an den Edge-Standort CloudFront kopiert. Änderungen an den ursprünglichen Inhalten werden zu den Edge-Standorten repliziert, indem die Besucher die Inhalte anfordern.

Verwendung CloudFront ohne Route 53

Das Tutorial auf dieser Seite verwendet Route 53, um auf Ihre CloudFront Verteilung hinzuweisen. Wenn Sie jedoch Inhalte bereitstellen möchten, die in einem Amazon S3-Bucket gehostet werden, CloudFront ohne Route 53 zu verwenden, finden Sie weitere Informationen unter [CloudFrontAmazon-Tutorials: Einrichtung einer dynamischen Inhaltsverteilung für Amazon S3](#). Wenn

Sie Inhalte bereitstellen, die in einem Amazon S3 S3-Bucket gehostet werden CloudFront, können Sie einen beliebigen Bucket-Namen verwenden, HTTP und beide HTTPS werden unterstützt.

Automatisieren der Einrichtung mit einem AWS CloudFormation Vorlage

Weitere Informationen zur Verwendung eines AWS CloudFormation Eine Vorlage für die Konfiguration einer sicheren statischen Website, die eine CloudFront Distribution für Ihre Website erstellt, finden Sie unter [Erste Schritte mit einer sicheren statischen Website](#) im Amazon CloudFront Developer Guide.

Themen

- [Schritt 1: Erstellen Sie eine CloudFront Distribution](#)
- [Schritt 2: Aktualisieren der Datensätze für Ihre Domäne und Unterdomäne](#)
- [\(Optional\) Schritt 3: Überprüfen der Protokolldateien](#)

Schritt 1: Erstellen Sie eine CloudFront Distribution

Zunächst erstellen Sie eine CloudFront Distribution. Damit steht Ihre Website in weltweit angesiedelten Rechenzentren zur Verfügung.

So erstellen Sie eine Verteilung mit einem Amazon S3-Ursprung

1. Öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Wählen Sie Create Distribution (Distribution erstellen).
3. Geben Sie auf der Seite Create Distribution (Verteilung erstellen) im Abschnitt Origin Settings (Ursprungseinstellungen) unter Origin Domain Name (Ursprungsdomänenname) den Amazon S3-Website-Endpunkt für Ihren Bucket ein – z. B. **example.com.s3-website.us-west-1.amazonaws.com**.


CloudFront füllt die Origin-ID für dich aus.

4. Behalten Sie für Default Cache Behavior Settings (Standardeinstellungen für das Zwischenspeicherverhalten) die Standardwerte bei.

Mit den Standardeinstellungen für die Viewer-Protokollrichtlinie kannst du sie HTTPS für deine statische Website verwenden. Weitere Informationen zu diesen Konfigurationsoptionen finden Sie unter [Werte, die Sie beim Erstellen oder Aktualisieren einer Web-Distribution angeben](#) im Amazon CloudFront Developer Guide.

5. Gehen Sie unter Distribution Settings (Verteilungseinstellungen) wie folgt vor:

- a. Lassen Sie für die Option Price Class (Preisklasse) die Einstellung Use All Edge Locations (Best Performance) (Alle Edge-Standorte verwenden (Beste Leistung)) unverändert.
- b. Legen Sie alternative Domainnamen (CNAMEs) für die Root-Domain und die www Subdomain fest. In diesem Tutorial sind dies `example.com` und `www.example.com`.

 **Important**

Bevor Sie diesen Schritt ausführen, beachten Sie die [Anforderungen für die Verwendung alternativer Domainnamen](#), insbesondere die Notwendigkeit eines gültigen SSL TLS /-Zertifikats.

- c. Wählen Sie für SSLZertifikat die Option Benutzerdefiniertes SSL Zertifikat (`example.com`) und wählen Sie das benutzerdefinierte Zertifikat aus, das die Domain- und Subdomainnamen abdeckt.

Weitere Informationen finden Sie unter [SSLZertifikat](#) im Amazon CloudFront Developer Guide.

- d. Geben Sie in Default Root Object (Standard-Root-Objekt) den Namen Ihres Indextdokuments ein, z. B. `index.html`.

Wenn die für den Zugriff auf die Distribution URL verwendete Datei keinen Dateinamen enthält, gibt die CloudFront Distribution das Indextdokument zurück. Das Default Root Object (Standard-Root-Objekt) sollte genau mit dem Namen des Indextdokuments für Ihre statische Website übereinstimmen. Weitere Informationen finden Sie unter [Konfigurieren eines Indextdokuments](#).

- e. Wählen Sie für Logging (Protokollierung) die Option On (Ein).

 **Important**

Wenn Sie eine Distribution erstellen oder aktualisieren und die CloudFront Protokollierung aktivieren, CloudFront aktualisiert die Bucket-Zugriffskontrollliste (ACL), sodass das `awslogsdelivery` Konto `FULL_CONTROL` berechtigt ist, Logs in Ihren Bucket zu schreiben. Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Konfiguration der Standardprotokollierung und für den Zugriff auf Ihre Protokolldateien](#) im Amazon CloudFront Developer Guide. Wenn der Bucket, der die Logs speichert, die vom Bucket-Besitzer erzwungene Einstellung zur Deaktivierung von S3 Object Ownership verwendet, CloudFront können keine

Logs in den Bucket geschrieben werden. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

- f. Wählen Sie als Bucket for Logs (Bucket für Protokolle) den Bucket zur Protokollierung aus, den Sie erstellt haben.

Weitere Informationen zum Konfigurieren eines Protokoll-Buckets finden Sie unter [\(Optional\) Protokollieren des Webdatenverkehrs](#).

- g. Wenn Sie die Protokolle, die durch den Datenverkehr zur CloudFront Distribution generiert werden, in einem Ordner speichern möchten, geben Sie im Feld Protokollpräfix den Ordernamen ein.
 - h. Behalten Sie für alle übrigen Einstellungen die Standardwerte bei.
6. Wählen Sie Create Distribution.
 7. Um den aktuellen Status der Verteilung anzuzeigen, suchen Sie die Verteilung in der Konsole, und prüfen Sie die Spalte Status.

Der Status InProgress gibt an, dass die Verteilung noch nicht vollständig bereitgestellt ist.

Nachdem Ihre Distribution bereitgestellt wurde, können Sie mit dem neuen CloudFront Domainnamen auf Ihre Inhalte verweisen.

8. Notieren Sie sich den Wert des Domainnamens, der in der CloudFront Konsole angezeigt wird, `dj4p1rv6mvubz.cloudfront.net` z. B.
9. Um zu überprüfen, ob Ihre CloudFront Distribution funktioniert, geben Sie den Domainnamen der Distribution in einen Webbrowser ein.

Wenn Ihre Website sichtbar ist, funktioniert die CloudFront Verteilung. Wenn auf Ihrer Website eine benutzerdefinierte Domain bei Amazon Route 53 registriert ist, benötigen Sie den CloudFront Domainnamen, um den Datensatz im nächsten Schritt zu aktualisieren.

Schritt 2: Aktualisieren der Datensätze für Ihre Domäne und Unterdomäne

Nachdem Sie nun erfolgreich eine CloudFront Verteilung erstellt haben, aktualisieren Sie den Aliaseintrag in Route 53 so, dass er auf die neue CloudFront Verteilung verweist.

Um den Aliaseintrag so zu aktualisieren, dass er auf eine CloudFront Verteilung verweist

1. Öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.

2. Wählen Sie in der linken Navigation Hosted Zones (Gehostete Zonen).
3. Wählen Sie auf der Seite Hosted Zones (Gehostete Zonen) die gehostete Zone aus, die Sie für Ihre Unterdomäne erstellt haben, beispielsweise `www.example.com`.
4. Wählen Sie unter Datensätzen den A-Eintrag aus, den Sie für Ihre Unterdomäne erstellt haben.
5. Wählen Sie unter Datensatzdetails den Befehl Datensatz bearbeiten.
6. Wählen Sie unter Traffic weiterleiten an die Option Alias to CloudFront Distribution aus.
7. Wählen Sie unter Verteilung auswählen die CloudFront Verteilung aus.
8. Wählen Sie Save (Speichern) aus.
9. Um den A-Eintrag für die Stammdomäne zur CloudFront Distribution umzuleiten, wiederholen Sie dieses Verfahren für die Stammdomäne, `example.com` z. B.

Die Aktualisierung der Datensätze wird innerhalb von 2 bis 48 Stunden wirksam.

10. Um zu sehen, ob die neuen A-Einträge wirksam geworden sind, geben Sie in einem Webbrowser Ihre Subdomain ein URL, z. B. `http://www.example.com`

Wenn der Browser Sie nicht mehr zur Stammdomäne umleitet (z. B. `http://example.com`), sind die neuen A-Datensätze vorhanden. Wenn der neue A-Eintrag wirksam geworden ist, wird der Datenverkehr, der vom neuen A-Eintrag an die CloudFront Distribution weitergeleitet wird, nicht an die Stammdomain umgeleitet. Alle Besucher, die über den nächstgelegenen CloudFront Standort auf die Website verweisen `http://example.com` oder zu diesem weitergeleitet `http://www.example.com` werden, profitieren von kürzeren Download-Zeiten.

Tip

Umleitungseinstellungen können von Browsern zwischengespeichert werden. Wenn Sie annehmen, dass die neuen A-Datensatzeinstellungen wirksam sind, aber trotzdem eine Umleitung von `http://www.example.com` nach `http://example.com` feststellen, löschen Sie zum Testen den Verlauf und den Cache des Browsers. Schließen Sie den Browser und öffnen Sie ihn wieder oder verwenden Sie einen anderen Webbrowser, wenn Sie einen weiteren installiert haben.

(Optional) Schritt 3: Überprüfen der Protokolldateien

Die Zugriffsprotokolle teilen Ihnen mit, wie viele Menschen die Website besuchen. Sie enthalten auch wertvolle Geschäftsdaten, die Sie mit anderen Diensten wie [Amazon](#) analysieren könnenEMR.

CloudFront Protokolle werden in dem Bucket und Ordner gespeichert, den Sie auswählen, wenn Sie eine CloudFront Distribution erstellen und die Protokollierung aktivieren. CloudFront schreibt innerhalb von 24 Stunden, nachdem die entsprechenden Anfragen gestellt wurden, Logs in Ihren Log-Bucket.

Die Protokolldateien für Ihre Website anzeigen

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Protokoll-Bucket für Ihre Website aus.
3. Wählen Sie den CloudFront Logs-Ordner aus.
4. Laden Sie die von geschriebenen .gzip Dateien herunter, CloudFront bevor Sie sie öffnen.

Wenn Sie die Website nur zur Übung erstellt haben, können Sie die von Ihnen zugewiesenen Ressourcen löschen, damit keine weiteren Kosten für Sie anfallen. Lesen Sie dazu den Abschnitt [Bereinigung Ihrer Beispielressourcen](#). Nachdem Sie Ihre gelöscht haben AWS Ressourcen, Ihre Website ist nicht mehr verfügbar.

Bereinigung Ihrer Beispielressourcen

Wenn Sie Ihre statische Website als Lernübung erstellt haben, sollten Sie die löschen AWS Ressourcen, die Sie so zugewiesen haben, dass Ihnen keine Gebühren mehr anfallen. Nachdem Sie Ihre gelöscht haben AWS Ressourcen, Ihre Website ist nicht mehr verfügbar.

Aufgaben

- [Schritt 1: Löschen Sie die CloudFront Amazon-Distribution](#)
- [Schritt 2: Löschen der von Route 53 gehosteten Zone](#)
- [Schritt 3: Deaktivieren der Protokollierung und Löschen Ihres S3-Buckets](#)

Schritt 1: Löschen Sie die CloudFront Amazon-Distribution

Bevor Sie eine CloudFront Amazon-Distribution löschen, müssen Sie sie deaktivieren. Eine deaktivierte Verteilung funktioniert nicht mehr und es fallen keine weiteren Kosten für sie an. Sie können eine deaktivierte Verteilung jederzeit wieder aktivieren. Nachdem Sie eine deaktivierte Verteilung gelöscht haben, ist sie nicht länger verfügbar.

Um eine CloudFront Distribution zu deaktivieren und zu löschen

1. Öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Klicken Sie mit der rechten Maustaste auf die Verteilung, die Sie deaktivieren möchten, und anschließend auf Disable (Deaktivieren).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Disable (Ja, deaktivieren).
4. Wählen Sie die deaktivierte Verteilung aus, und klicken Sie dann auf Delete (Löschen).
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete.

Schritt 2: Löschen der von Route 53 gehosteten Zone

Bevor Sie eine gehostete Zone löschen können, müssen Sie die zuvor erstellten Datensätze löschen. Sie müssen den NS und die SOA Datensätze nicht löschen. Diese werden automatisch gelöscht, wenn Sie die gehostete Zone löschen.

Sie löschen die Datensätze wie folgt:

1. Öffnen Sie die Route 53-Konsole unter <https://console.aws.amazon.com/route53/>.
2. Wählen Sie in der Liste der Domännennamen Ihren Domännennamenaus, und wählen Sie anschließend Go to Record Sets (Zu den Datensätzen).
3. Wählen Sie in der Liste der Datensatzsätze die A-Datensätze aus, die Sie erstellt haben.

Der einzelne Datensatztyp wird in der Spalte Type (Typ) aufgeführt.

4. Wählen Sie Delete Record Set (Datensatz löschen).
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Confirm (Bestätigen).

So löschen Sie eine von Route 53 gehostete Zone:

1. Setzen Sie das vorherige Verfahren fort, indem Sie Back to Hosted Zones (Zurück zu gehosteten Zonen) auswählen.
2. Wählen Sie Ihren Domännennamen und dann Delete Hosted Zone (Gehostete Zone löschen).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Confirm (Bestätigen).

Schritt 3: Deaktivieren der Protokollierung und Löschen Ihres S3-Buckets

Bevor Sie Ihren S3-Bucket löschen, stellen Sie sicher, dass die Protokollierung für den Bucket deaktiviert ist. Andernfalls AWS schreibt weiterhin Logs in Ihren Bucket, während Sie ihn löschen.

Deaktivieren Sie die Protokollierung für einen Bucket wie folgt:

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie unter Buckets (Buckets) den Namen Ihres Buckets aus. Wählen Sie anschließend Properties (Eigenschaften) aus.
3. Wählen Sie unter Properties (Eigenschaften) Logging (Protokollierung) aus.
4. Deaktivieren Sie das Kontrollkästchen Enabled (Aktiviert).
5. Wählen Sie Save (Speichern) aus.

Jetzt können Sie Ihren Bucket löschen. Weitere Informationen finden Sie unter [Löschen eines Buckets](#).

Erstellen, Konfigurieren und Arbeiten mit Amazon S3-Buckets

Um Ihre Daten in Amazon S3 zu speichern, arbeiten Sie mit Ressourcen, die als Buckets und Objekte bezeichnet werden. Ein Bucket ist ein Container für Objekte. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben.

Um ein Objekt in Amazon S3 zu speichern, erstellen Sie einen Bucket und laden das Objekt dann in einen Bucket hoch. Wenn sich das Objekt im Bucket befindet, können Sie es öffnen, herunterladen und verschieben. Wenn Sie kein Objekt oder einen Bucket mehr benötigen, können Sie Ihre Ressourcen aufräumen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Note

Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon S3](#). Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter [AWS Kostenloses Kontingent](#).

Die Themen in diesem Abschnitt geben einen Überblick über die Arbeit mit Buckets in Amazon S3. Sie enthalten Informationen über das Benennen, Erstellen, Zugreifen und Löschen von Buckets. Weitere Informationen zum Anzeigen oder Auflisten von Objekten in einem Bucket finden Sie unter [Organisieren, Auflisten und Arbeiten mit Ihren Objekten](#).

Themen

- [Bucket-Übersicht](#)
- [Regeln für die Benennung von Buckets](#)
- [Zugreifen auf einen Amazon-S3-Bucket und Auflisten des Buckets](#)

- [Erstellen eines Buckets](#)
- [Anzeigen der Eigenschaften eines S3-Buckets](#)
- [Leeren eines Buckets](#)
- [Löschen eines Buckets](#)
- [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#)
- [Arbeiten mit Mountpoint für Amazon S3](#)
- [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#)
- [Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung](#)
- [Bucket-Kontingente, Einschränkungen und Beschränkungen](#)

Bucket-Übersicht

Um Ihre Daten (Fotos, Videos, Dokumente usw.) auf Amazon S3 hochzuladen, müssen Sie zunächst einen S3-Bucket in einem der AWS-Regionen.

Ein Bucket ist ein Behälter für Objekte, die in Amazon S3 gespeichert werden. Sie können beliebig viele Objekte in einem Bucket speichern und bis zu 100 Buckets in Ihrem Konto haben. Um eine Erhöhung anzufordern, rufen Sie die [Service-Quotas-Konsole](#) auf.

Jedes Objekt ist in einem Bucket enthalten. Wenn das benannte Objekt beispielsweise im amzn-s3-demo-bucket Bucket in der Region USA West (Oregon) gespeichert photos/puppy.jpg ist, kann es mit dem URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg` adressiert werden. Weitere Informationen siehe [Zugriff auf einen Bucket](#).

Hinsichtlich der Implementierung sind Buckets und Objekte AWS Ressourcen, und Amazon S3 bietet Ihnen APIs die Möglichkeit, diese zu verwalten. Sie können beispielsweise mit Amazon S3 einen Bucket erstellen und Objekte hochladenAPI. Sie können diese Vorgänge auch in der Amazon-S3-Konsole durchführen. Die Konsole verwendet Amazon S3APIs, um Anfragen an Amazon S3 zu senden.

In diesem Abschnitt wird beschrieben, wie Buckets verwendet werden. Weitere Informationen zur Arbeit mit Objekten finden Sie unter [Übersicht über Amazon-S3-Objekte](#).

Amazon S3 unterstützt globale Buckets, was bedeutet, dass jeder Bucket-Name für alle eindeutig sein muss AWS-Konten in allen AWS-Regionen innerhalb einer Partition. Eine Partition ist eine Gruppe von Regionen. AWS hat derzeit drei Partitionen: aws (Standardregionen), aws-cn (China-Regionen) und aws-us-gov (AWS GovCloud (US)).

Nachdem ein Bucket erstellt wurde, kann der Name dieses Buckets nicht von einem anderen verwendet werden AWS-Konto in derselben Partition, bis der Bucket gelöscht wird. Verlassen Sie sich nicht auf eine spezifische Benennungskonvention für Buckets für Verfügbarkeits- oder Sicherheitsprüfungszwecke. Namensrichtlinien für Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

Amazon S3 erstellt Buckets in der von Ihnen angegebenen Region. Um die Latenz zu reduzieren, die Kosten zu minimieren oder regulatorische Anforderungen zu erfüllen, wählen Sie eine beliebige AWS-Region das sich geografisch in Ihrer Nähe befindet. Wenn Sie beispielsweise in Europa ansässig sind, könnte es vorteilhaft sein, Buckets in den Regionen EU (Irland) oder EU (Frankfurt) zu erstellen. Eine Liste der Amazon S3 S3-Regionen finden Sie unter [Regionen und Endpunkte](#) in der AWS Allgemeine Referenz.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Note

Objekte, die zu einem Bucket gehören, den Sie in einem bestimmten Bucket erstellen AWS-Region verlassen Sie diese Region niemals, es sei denn, Sie übertragen sie ausdrücklich in eine andere Region. Beispielsweise verlassen in der Region EU (Irland) gespeicherte Objekte diese Region nie.

Themen

- [Berechtigungen](#)
- [Verwalten des öffentlichen Zugriffs auf Buckets](#)
- [Optionen für die Bucket-Konfiguration](#)

Berechtigungen

Du kannst deine verwenden Root-Benutzer des AWS-Kontos Anmeldeinformationen, um einen Bucket zu erstellen und jeden anderen Amazon S3 S3-Vorgang auszuführen. Wir empfehlen jedoch,

dass Sie nicht die Root-Benutzeranmeldedaten Ihres AWS-Konto um Anfragen zu stellen, z. B. um einen Bucket zu erstellen. Erstellen Sie stattdessen ein AWS Identity and Access Management (IAM) Benutzer und gewähren Sie diesem Benutzer vollen Zugriff (Benutzer haben standardmäßig keine Berechtigungen).

Diese Benutzer werden als Administratoren bezeichnet. Sie können die Administrator-Benutzeranmeldedaten anstelle der Root-Benutzeranmeldedaten Ihres Kontos verwenden, um mit AWS und Aufgaben auszuführen, z. B. einen Bucket erstellen, Benutzer erstellen und ihnen Berechtigungen erteilen.

Weitere Informationen finden Sie unter [Root-Benutzer des AWS-Kontos Anmeldeinformationen und IAM Benutzeranmeldedaten](#) in AWS Allgemeine Informationen und [bewährte Sicherheitsmethoden finden Sie IAM im IAM Benutzerhandbuch](#).

Das Tool AWS-Konto das eine Ressource erstellt, besitzt diese Ressource. Zum Beispiel, wenn Sie einen IAM Benutzer in Ihrem AWS-Konto und dem Benutzer die Erlaubnis erteilen, einen Bucket zu erstellen, kann der Benutzer einen Bucket erstellen. Aber der Bucket gehört nicht dem Benutzer; der AWS-Konto dem der Benutzer angehört, gehört der Bucket. Der Benutzer benötigt eine zusätzliche Berechtigung vom Ressourceneigentümer, um andere Bucket-Operation auszuführen. Weitere Informationen zum Verwalten von Berechtigungen für Ihre Amazon-S3-Ressourcen finden Sie unter [Identity and Access Management für Amazon S3](#).

Verwalten des öffentlichen Zugriffs auf Buckets

Öffentlicher Zugriff auf Buckets und Objekte wird über Bucket-Richtlinien, Zugriffskontrolllisten (ACLs) oder beides gewährt. Um Sie bei der Verwaltung des öffentlichen Zugriffs auf Amazon-S3-Ressourcen zu unterstützen, bietet Amazon S3 Einstellungen für das Sperren des öffentlichen Zugriffs. Die Einstellungen von Amazon S3 Block Public Access können Richtlinien außer Kraft setzen ACLs und so einheitliche Beschränkungen für den öffentlichen Zugriff auf diese Ressourcen durchsetzen. Sie können Block Public Access-Einstellungen auf einzelne Buckets oder auf alle Buckets in Ihrem Konto anwenden.

Um zu gewährleisten, dass bei allen Amazon-S3-Buckets und -Objekten der öffentliche Zugriff blockiert ist, sind alle vier Einstellungen für Block Public Access standardmäßig aktiviert, wenn Sie einen neuen Bucket erstellen. Wir empfehlen, alle vier Einstellungen für Block Public Access auch für Ihr Konto zu aktivieren. Diese Einstellungen blockieren den öffentlichen Zugriff für alle aktuellen und künftigen Buckets.

Bevor Sie diese Einstellungen anwenden, verifizieren Sie, dass Ihre Anwendungen ohne öffentlichen Zugriff korrekt funktionieren. Wenn ein bestimmtes Maß an öffentlichem Zugriff auf Ihre Buckets oder Objekte nötig ist, z. B. zum Hosten einer statischen Website, wie unter [Hosten einer statischen Website mit Amazon S3](#) beschrieben, können Sie die einzelnen Einstellungen an Ihre Speicheranwendungsfälle anpassen. Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Wir empfehlen jedoch dringend, Block Public Access aktiviert zu lassen. Wenn Sie alle vier Block Public Access-Einstellungen aktiviert lassen und eine statische Website hosten möchten, können Sie Amazon CloudFront Origin Access Control (OAC) verwenden. Amazon CloudFront bietet die Funktionen, die für die Einrichtung einer sicheren statischen Website erforderlich sind. Statische Amazon S3 S3-Websites unterstützen nur HTTP Endpunkte. Amazon CloudFront verwendet den dauerhaften Speicher von Amazon S3 und bietet gleichzeitig zusätzliche Sicherheitsheader, wie HTTPS z. HTTPS erhöht die Sicherheit, indem es eine normale HTTP Anfrage verschlüsselt und vor häufigen Cyberangriffen schützt.

Weitere Informationen finden Sie unter [Erste Schritte mit einer sicheren statischen Website](#) im Amazon CloudFront Developer Guide.

Note

Wenn beim Auflisten Ihrer Buckets mit ihren Einstellungen für den öffentlichen Zugriff die Fehlermeldung `Error` angezeigt wird, verfügen Sie möglicherweise nicht über die erforderlichen Berechtigungen. Stellen Sie sicher, dass Sie Ihrer Benutzer- oder Rollenrichtlinie die folgenden Berechtigungen hinzugefügt haben:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3:ListAccessPoints
s3:ListAllMyBuckets
```

In einigen seltenen Fällen können Anfragen auch aufgrund eines fehlschlagen AWS-Region Ausfall.

Optionen für die Bucket-Konfiguration

Amazon S3 unterstützt Vorgänge Optionen für die Konfiguration Ihres Buckets. Sie können Ihren Bucket beispielsweise für ein Website-Hosting konfigurieren, eine Konfiguration zur Verwaltung des Lebenszyklus von Objekten im Bucket hinzufügen und den Bucket so konfigurieren, dass alle Zugriffe protokolliert werden. Amazon S3 unterstützt Subressourcen, um die Bucket-Konfigurationsinformationen zu speichern und zu verwalten. Sie können Amazon S3 verwendenAPI, um diese Unterressourcen zu erstellen und zu verwalten. Sie können jedoch auch die Konsole oder die AWS SDKs.

Note

Außerdem gibt es Konfigurationen auf Objektebene. Sie können beispielsweise Berechtigungen auf Objektebene konfigurieren, indem Sie eine spezifische Zugriffskontrollliste (ACL) für dieses Objekt konfigurieren.

Man bezeichnet sie als Subressourcen, weil sie im Kontext eines spezifischen Buckets oder Objekts existieren. Die folgende Tabelle listet Subressourcen auf, die ihnen ermöglichen, für den Bucket spezifische Konfigurationen zu verwalten.

Subressource	Beschreibung
cors (ursprung übergreifende gemeinsame Nutzung von Ressourcen)	Sie können Ihren Bucket so konfigurieren, dass er ursprungsübergreifende Anforderungen zulässt. Weitere Informationen finden Sie unter Verwendung von ursprungsübergreifendem Ressourcenaustausch () CORS .
event notification	Sie können Ihrem Bucket gestatten, Benachrichtigungen über bestimmte Bucket-Ereignisse zu senden. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen .
Lebenszyklus	Sie können Lebenszyklusregeln für Objekte in Ihrem Bucket mit definiertem Lebenszyklus definieren. Sie können z. B. eine Regel so festlegen, dass

Subressource	Beschreibung
	<p>Objekte ein Jahr nach dem Erstellungsdatum archiviert werden, oder dass ein Objekt 10 Jahre nach dem Erstellen gelöscht wird.</p> <p>Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.</p>
location	<p>Wenn Sie einen Bucket erstellen, geben Sie Folgendes an AWS-Region wo Amazon S3 den Bucket erstellen soll. Amazon S3 speichert diese Informationen in der Standort-Subressource und bietet Ihnen die MöglichkeitAPI, diese Informationen abzurufen.</p>
logging	<p>Protokollierung ermöglicht Ihnen, Zugriffsanforderungen für Ihren Bucket nachzuverfolgen. Jeder Zugriffsprotokolldatensatz enthält Details über eine Zugriffsanforderung, z. B. Auftraggeber, Bucket-Name, Anforderungszeit, Anforderungsaktion, Antwortstatus und Fehlercode, falls vorhanden. Die Zugriffsprotokollinformationen können für Sicherheits- und Zugriffsüberprüfungen nützlich sein. Außerdem erfahren Sie damit mehr über Ihren Kundenstamm und erhalten einen Überblick über Ihre Amazon-S3-Rechnung.</p> <p>Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.</p>
Objektsperre	<p>Zur Verwendung der S3-Objektsperre muss diese für einen Bucket aktiviert werden. Sie können optional auch einen Standardaufbewahrungsmodus und -zeitraum konfigurieren, der für neue Objekte gilt, die in den Bucket platziert werden.</p> <p>Weitere Informationen finden Sie unter Verwenden der S3-Objektsperre.</p>
Richtlinie und ACL(Zugriffs kontrollliste)	<p>Standardmäßig sind alle Ihre Ressourcen (wie Buckets und Objekte) privat. Amazon S3 unterstützt sowohl die Optionen Bucket Policy als auch Access Control List (ACL), mit denen Sie Berechtigungen auf Bucket-Ebene gewähren und verwalten können. Amazon S3 speichert die Berechtigungsinformationen in den Subressourcen policy und acl.</p> <p>Weitere Informationen finden Sie unter Identity and Access Management für Amazon S3.</p>

Subressource	Beschreibung
Replikation	Replikation ist das automatische, asynchrone Kopieren von Objekten zwischen Buckets in verschiedenen oder denselben AWS-Regionen. Weitere Informationen finden Sie unter Überblick über das Replizieren von Objekten .
requestPayment	<p>Standardmäßig ist der AWS-Konto der den Bucket erstellt (der Bucket-Besitzer) bezahlt für Downloads aus dem Bucket. Mit dieser Subressource kann der Bucket-Eigentümer angeben, dass die Person, die den Download anfordert, die Gebühren für den Download trägt. Amazon S3 bietet Ihnen eine API Möglichkeit, diese Unterressource zu verwalten.</p> <p>Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung.</p>
Markieren	<p>Sie können Ihrem Bucket Tags zur Kostenzuweisung hinzufügen, um Ihre Kosten zu kategorisieren und nachzuverfolgen AWS Kosten. Amazon S3 unterstützt die tagging-Subressource, um Markierungen für einen Bucket zu speichern und zu verwalten. Mithilfe von Tags, die Sie auf Ihren Bucket anwenden, AWS generiert einen Kostenverteilungsbericht mit Nutzung und Kosten, die nach Ihren Stichwörtern zusammengefasst sind.</p> <p>Weitere Informationen finden Sie unter Fakturierung und Nutzungsberichte für Amazon S3.</p>
transfer acceleration	<p>Transfer Acceleration ermöglicht schnelle, einfache und sichere Übertragungen von Dateien zwischen Ihrem Client und einem S3-Bucket über große Entfernungen. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte von Amazon CloudFront.</p> <p>Weitere Informationen finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.</p>

Subressource	Beschreibung
Versioning	<p>Versioning hilft Ihnen bei einer Wiederherstellung nach einem versehentlichen Überschreiben und Löschen.</p> <p>Wir empfehlen das Versioning als bewährte Methode, um zu verhindern, dass Objekte versehentlich gelöscht oder überschrieben werden.</p> <p>Weitere Informationen finden Sie unter Verwenden der Versioning in S3-Buckets.</p>
Website	<p>Sie können Ihren Bucket für ein Hosting statischer Websites konfigurieren. Amazon S3 speichert diese Konfiguration, indem es eine website-Subressource erstellt.</p> <p>Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.</p>

Regeln für die Benennung von Buckets

Die folgenden Regeln gelten für die Benennung von Buckets für allgemeine Zwecke und von Verzeichnis-Buckets in Amazon S3:

Themen

- [Regeln für die Benennung von Buckets für allgemeine Zwecke](#)
- [Regeln für die Benennung von Verzeichnis-Buckets](#)

Regeln für die Benennung von Buckets für allgemeine Zwecke


Die folgenden Regeln gelten für die Benennung von Buckets für allgemeine Zwecke.

- Bucket-Namen müssen zwischen (min.) 3 und (max.) 63 Zeichen lang sein.
- Bucket-Namen können nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen.
- Bucket-Namen müssen mit einem Buchstaben oder einer Zahl beginnen und enden.
- Bucketnamen dürfen keine aufeinander folgenden Punkte (..) enthalten.

- Bucket-Namen dürfen nicht als IP-Adresse formatiert sein (zum Beispiel 192.168.5.4).
- Der Bucket-Name darf nicht mit dem Präfix `xn-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
- Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#).
- Bucket-Namen dürfen nicht mit dem Suffix `--o1-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets](#).
- Bucket-Namen dürfen nicht mit dem Suffix `.mrap` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter [Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten](#).
- Bucket-Namen dürfen nicht mit dem Suffix `--x-s3` enden. Dieses Suffix ist für Verzeichnis-Buckets reserviert. Weitere Informationen finden Sie unter [Regeln für die Benennung von Verzeichnis-Buckets](#).
- Bucket-Namen müssen für alle eindeutig sein AWS-Konten in allen AWS-Regionen innerhalb einer Partition. Eine Partition ist eine Gruppe von Regionen. AWS hat derzeit drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US)).
- Ein Bucket-Name kann nicht von einem anderen verwendet werden AWS-Konto in derselben Partition, bis der Bucket gelöscht wird.
- Buckets, die mit Amazon S3 Transfer Acceleration verwendet werden, können keine Punkte (.) in ihren Namen haben. Weitere Informationen zu Transfer Acceleration finden Sie unter [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#).

Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden. Wenn Sie Punkte in den Namen eines Buckets aufnehmen, können Sie virtual-host-style Addressing Over nicht verwenden HTTPS, es sei denn, Sie führen Ihre eigene Zertifikatsvalidierung durch. Dies liegt daran, dass die Sicherheitszertifikate, die für das virtuelle Hosten von Buckets verwendet werden, nicht für Buckets mit Punkten in ihren Namen funktionieren.

Diese Einschränkung wirkt sich nicht auf Buckets aus, die für statisches Website-Hosting verwendet werden, da statisches Website-Hosting nur über HTTP verfügbar ist. Weitere Informationen zur virtual-host-style Adressierung finden Sie unter [Virtuelles Hosting bei Buckets](#). Weitere Hinweise zum Hosten statischer Websites finden Sie unter [Hosten einer statischen Website mit Amazon S3](#).

 Note

Vor dem 1. März 2018 konnten Buckets, die in der Region USA Ost (Nord-Virginia) erstellt wurden, Namen mit bis zu 255 Zeichen und mit Großbuchstaben und Unterstrichen haben. Ab dem 1. März 2018 müssen neue Buckets in USA Ost (Nord-Virginia) den gleichen Regeln entsprechen, die in allen anderen Regionen angewendet werden.

Informationen zu Objektschlüsselnamen finden Sie unter [Erstellen von Objektschlüsselnamen](#).

Beispiel für Namen von Buckets für allgemeine Zwecke

Dies sind Beispiele für gültige Namen von Buckets. Sie befolgen die Empfehlungen für die Benennung von Buckets für allgemeine Zwecke:

- docexamplebucket1
- log-delivery-march-2020
- my-hosted-content

Die folgenden Beispiel-Bucket-Namen sind gültig, aber nicht für andere Verwendungszwecke als statisches Website-Hosting empfohlen:

- docexamplewebsite.com
- www.docexamplewebsite.com
- my.example.s3.bucket

Die folgenden Beispiel-Bucket-Namen sind ungültig:

- doc_example_bucket (enthält Unterstriche)
- DocExampleBucket (enthält Großbuchstaben)
- doc-example-bucket- (endet mit einem Bindestrich)

Regeln für die Benennung von Verzeichnis-Buckets

Die folgenden Benennungsregeln gelten für Verzeichnis-Buckets.

- Seien Sie innerhalb der Auserwählten einzigartig AWS-Region und Availability Zone.
- Der Name muss zwischen 3 (min) und 63 (maximal) Zeichen lang sein, einschließlich des Suffixes.
- Nur aus Kleinbuchstaben, Zahlen und Bindestrichen bestehen.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.
- Muss das folgende Suffix enthalten: `--azid--x-s3`
- Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
- Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#).
- Bucket-Namen dürfen nicht mit dem Suffix `--o1-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets](#).
- Bucket-Namen dürfen nicht mit dem Suffix `.mrp` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter [Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten](#).

Note

Wenn Sie mithilfe der Konsole einen Verzeichnis-Bucket erstellen, wird dem von Ihnen angegebenen Basisnamen automatisch ein Suffix hinzugefügt. Dieses Suffix enthält die ID der Availability Zone, die Sie ausgewählt haben.

Wenn Sie einen Verzeichnis-Bucket mithilfe von erstellen, müssen API Sie das vollständige Suffix, einschließlich der Availability Zone ID, in Ihrer Anfrage angeben. Eine Liste der Availability Zone finden Sie IDs unter [Availability Zones und Regionen bei S3 Express One Zone](#).

Zugreifen auf einen Amazon-S3-Bucket und Auflisten des Buckets

Sie können verschiedene Tools verwenden, um auf Amazon-S3-Buckets zuzugreifen und sie aufzulisten. Gehen Sie die folgenden Tools durch und finden Sie heraus, welcher Ansatz für Ihren Anwendungsfall geeignet ist:

- **Amazon-S3-Konsole:** Mit der Amazon-S3-Konsole können Sie problemlos auf einen Bucket zugreifen und seine Eigenschaften ändern. Mit der Benutzeroberfläche der Konsole können Sie zudem fast alle Bucket-Operationen ausführen, ohne Code schreiben zu müssen.
- **AWS CLI:** Wenn Sie auf mehrere Buckets zugreifen müssen, können Sie Zeit sparen, indem Sie AWS Command Line Interface (AWS CLI), um häufig auftretende und sich wiederholende Aufgaben zu automatisieren. Die Skriptfähigkeit und die Wiederholbarkeit gängiger Aktionen sind wichtige Aspekte, die bei der Skalierung von Unternehmen zu berücksichtigen sind. Weitere Informationen finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#).
- **Amazon S3 RESTAPI:** Sie können Amazon S3 verwenden, um Ihre eigenen Programme REST API zu schreiben und programmgesteuert auf Buckets zuzugreifen. Amazon S3 unterstützt eine API Architektur, in der Ihre Buckets und Objekte Ressourcen sind, von denen jede über eine Ressource verfügtURI, die die Ressource eindeutig identifiziert. Weitere Informationen finden Sie unter [Entwickeln mit Amazon S3 unter Verwendung der REST-API](#).

Je nach Anwendungsfall des Amazon-S3-Buckets werden verschiedene Methoden für den Zugriff auf die zugrunde liegenden Daten in den Buckets empfohlen. Die folgende Liste enthält gängige Anwendungsfälle für den Zugriff auf Daten.

- **Statische Websites** – Sie können in Amazon S3 eine statische Website hosten. Bei diesem Anwendungsfall lässt sich der S3-Bucket so konfigurieren, dass er sich wie eine Website verhält. Ein Beispiel für Schritte zum Hosten einer Website auf Amazon S3 finden Sie unter [Tutorial: Konfigurieren einer statischen Website auf Amazon S3](#).

Um eine statische Website mit aktivierten Sicherheitseinstellungen wie Block Public Access zu hosten, empfehlen wir, Amazon CloudFront mit Origin Access Control (OAC) zu verwenden und zusätzliche Sicherheitsheader zu implementieren, wie HTTPS z. Weitere Informationen finden Sie unter [Erste Schritte mit einer sicheren statischen Website](#).

Note

Amazon S3 unterstützt sowohl den [virtuellen Hosting-Stil als auch den URL-Pfadstil](#) für den statischen Zugriff auf Websites. Da auf Buckets im Pfadstil und im URLs Virtual-Hosted-Stil zugegriffen werden kann, empfehlen wir, Buckets mit -konformen Bucket-Namen zu erstellen. DNS Weitere Informationen finden Sie unter [Bucket-Kontingente, Einschränkungen und Beschränkungen](#).

- Freigegebene Datensätze – Bei der Skalierung auf Amazon S3 wird üblicherweise ein Multi-Tenant-Modell verwendet, bei dem Sie verschiedenen Endkunden oder Geschäftseinheiten eindeutige Präfixe innerhalb eines freigegebenen Buckets zuweisen. Durch die Verwendung von [Amazon S3 Access Points](#) können Sie eine umfangreiche Bucket-Richtlinie in separate, diskrete Zugangspunkt-Richtlinien für jede Anwendung unterteilen, die auf den freigegebenen Datensatz zugreifen muss. Mit diesem Ansatz ist es einfacher, sich auf die Erstellung der richtigen Zugriffsrichtlinie für eine Anwendung zu konzentrieren, ohne die Aktivitäten anderer Anwendungen innerhalb des freigegebenen Datensatzes zu beeinflussen. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).
- Workload mit hohem Durchsatz – Mountpoint für Amazon S3 ist ein Open-Source-Dateiclient mit hohem Durchsatz zum Mounten eines Amazon-S3-Bucket als lokales Dateisystem. Mit Mountpoint können Anwendungen über Dateisystem-Operationen wie Öffnen und Lesen auf Objekte zugreifen, die in Amazon S3 gespeichert sind. Mountpoint übersetzt diese Operationen automatisch in API S3-Objekt-Aufrufe, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und Durchsatz von Amazon S3 zugreifen können. Weitere Informationen finden Sie unter [Arbeiten mit Mountpoint für Amazon S3](#).
- Multi-Region-Anwendungen — Amazon S3 Multi-Region Access Points bieten einen globalen Endpunkt, über den Anwendungen Anfragen von S3-Buckets bearbeiten können, die sich in mehreren AWS-Regionen. Sie können Multi-Region-Access Points verwenden, um multiregionale Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Anstatt Anforderungen über das öffentliche Internet zu senden, bieten Multi-Region Access Points integrierte Netzwerkausfallsicherheit mit Beschleunigung internetbasierter Anforderungen an Amazon S3. Weitere Informationen finden Sie unter [Multi-Regions-Zugriffspunkte in Amazon S3](#).
- Neue Anwendungen erstellen — Sie können das verwenden AWS SDKs bei der Entwicklung von Anwendungen mit Amazon S3. Das Tool AWS SDKs vereinfachen Sie Ihre Programmieraufgaben, indem Sie das zugrunde liegende Amazon S3 einschließen RESTAPI. Um miteinander verbundene

Mobil- und Webanwendungen zu erstellen, können Sie die AWS Mobil SDKs und das AWS Amplify JavaScript Bibliothek. Weitere Informationen finden Sie unter [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

- Secure Shell (SSH) File Transfer Protocol (SFTP) — Wenn Sie versuchen, vertrauliche Daten sicher über das Internet zu übertragen, können Sie einen SFTP -fähigen Server mit Ihrem Amazon S3 S3-Bucket verwenden. AWS SFTP ist ein Netzwerkprotokoll, das die volle Sicherheits- und Authentifizierungsfunktionalität von SSH unterstützt. Mit diesem Protokoll haben Sie eine genaue Kontrolle über Benutzeridentität, Berechtigungen und Schlüssel oder Sie können IAM Richtlinien verwenden, um den Zugriff zu verwalten. Um Ihrem Amazon S3 S3-Bucket einen SFTP aktivierten Server zuzuordnen, stellen Sie sicher, dass Sie zuerst Ihren SFTP -fähigen Server erstellen. Anschließend richten Sie Benutzerkonten ein und verknüpfen den Server mit einem Amazon-S3-Bucket. Eine exemplarische Vorgehensweise für diesen Prozess finden Sie unter [AWS Transfer for SFTP — Vollständig verwalteter SFTP Service für Amazon S3](#) in AWS Blogs.

Auflisten eines Buckets

Zum Auflisten sämtlicher Buckets benötigen Sie die Genehmigung `s3:ListAllMyBuckets`. Um auf einen Bucket zuzugreifen, stellen Sie sicher, dass Sie auch den erforderlichen Bucket erhalten AWS Identity and Access Management (IAM) Berechtigungen zum Auflisten des Inhalts des angegebenen Buckets. Ein Beispiel für eine Bucket-Richtlinie, die den Zugriff auf einen S3-Bucket gewährt, finden Sie unter [Einem IAM Benutzer Zugriff auf einen Ihrer Buckets gewähren](#). Wenn Sie auf den Fehler HTTP Zugriff verweigert (403 Forbidden) stoßen, finden Sie weitere Informationen unter [Bucket-Richtlinien und IAM Richtlinien](#).

Sie können Ihren Bucket mithilfe der Amazon S3 S3-Konsole auflisten, AWS CLI, oder das AWS SDKs.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie aus der Liste der Buckets für allgemeine Zwecke den Bucket aus, den Sie anzeigen möchten.

Note

Die Liste der Buckets für allgemeine Zwecke enthält Buckets, die sich in allen Bereichen befinden AWS-Regionen.

Verwendung der AWS CLI

Um das AWS CLI Verwenden Sie den Befehl, um auf einen S3-Bucket zuzugreifen oder eine Liste von S3-Buckets zu generieren. `ls` Sollen alle Objekte im Bucket aufgelistet werden, müssen Sie über die Berechtigung `s3:ListBucket` verfügen.

Um diesen Beispielbefehl zu verwenden, ersetzen *DOC-EXAMPLE-BUCKET1* mit dem Namen Ihres Buckets.

```
$ aws s3 ls s3://DOC-EXAMPLE-BUCKET1
```

Mit dem folgenden Beispielbefehl werden alle Amazon-S3-Buckets in Ihrem Konto aufgelistet.

```
$ aws s3 ls
```

Weitere Informationen finden Sie unter [Auflisten von Buckets und Objekten](#).

Verwendung der AWS SDKs

Sie können auch auf einen Amazon S3 S3-Bucket zugreifen, indem Sie den [ListBuckets](#)APIBetrieb. Beispiele für die Verwendung dieser Operation mit verschiedenen AWS SDKs, siehe [Verwenden Sie ListBuckets mit einem AWS SDKoder CLI](#).

Erstellen eines Buckets

Um Ihre Daten auf Amazon S3 hochzuladen, müssen Sie zunächst einen Amazon S3 S3-Bucket in einem der AWS-Regionen. Wenn Sie einen Bucket erstellen, müssen Sie einen Bucket-Namen und eine Region auswählen. Sie können optional andere Speicherverwaltungsoptionen für den Bucket auswählen. Der Name eines erstellten Buckets oder Region kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

Das Tool AWS-Konto der den Bucket erstellt, gehört ihm. Sie können beliebig viele Objekte in den Bucket hochladen. Standardmäßig können Sie in jedem Ihrer Buckets bis zu 100 Buckets erstellen AWS-Konten. Wenn Sie mehr Buckets benötigen, können Sie das Bucketlimit Ihres Kontos auf maximal 1.000 Buckets erhöhen, indem Sie eine Erhöhung des Service-Limits einreichen. Informationen dazu, wie Sie eine Erhöhung des Bucket-Limits einreichen können, finden Sie unter [AWS-Service Kontingente](#) in der AWS Allgemeine Referenz. Sie können in einem Bucket beliebig viele Objekte speichern.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um Zugriffskontrolllisten zu deaktivieren oder zu aktivieren (ACLs). Standardmäßig ist für Object Ownership die erzwungene Einstellung Bucket Owner festgelegt, und alle sind ACLs deaktiviert. Wenn diese ACLs Option deaktiviert ist, besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Richtlinien.

Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Basisebene der Verschlüsselungskonfiguration für jeden Bucket in Amazon S3. Alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden automatisch mit SSE-S3 als Basisverschlüsselungseinstellung verschlüsselt. Wenn Sie eine andere Art der Standardverschlüsselung verwenden möchten, können Sie auch eine serverseitige Verschlüsselung mit angeben AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder vom Kunden bereitgestellte Schlüssel (SSE-C) zur Verschlüsselung Ihrer Daten. Weitere Informationen finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

Sie können die Amazon S3-Konsole verwenden, Amazon S3 APIs, AWS CLI, oder AWS SDK um einen Bucket zu erstellen. Weitere Informationen zu den Berechtigungen, die zum Erstellen eines Buckets erforderlich sind, finden Sie [CreateBucket](#) in der Amazon Simple Storage Service API Reference.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

 Note

Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Für eine Liste von Amazon S3 AWS-Regionen, siehe [AWS-Service Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

Anschließend wird die Seite Bucket erstellen geöffnet.

5. Sehen Sie sich unter Allgemeine Konfiguration die AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Allgemein aus.
7. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

Der Bucket-Name ...:

- Muss innerhalb einer Partition einzigartig sein. Eine Partition ist eine Gruppe von Regionen. AWS hat derzeit drei Partitionen: aws (Standardregionen), aws-cn (Regionen in China) und aws-us-gov (AWS GovCloud (US) Regions).
- zwischen 3 und 63 Zeichen lang sein,
- Darf nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen. Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Das Tool AWS-Konto wer den Bucket erstellt, besitzt ihn. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

⚠ Important

Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name ist in dem Bereich sichtbarURLs, der auf die Objekte im Bucket verweist.

8. AWS Management Console ermöglicht es Ihnen, die Einstellungen eines vorhandenen Buckets in Ihren neuen Bucket zu kopieren. Wenn Sie die Einstellungen eines vorhandenen Buckets nicht kopieren möchten, fahren Sie mit dem nächsten Schritt fort.

ℹ Note

Diese Option:

- Ist nicht verfügbar in der AWS CLI und ist nur in der Konsole verfügbar
- Ist für Verzeichnis-Buckets nicht verfügbar
- Kopiert die Bucket-Richtlinie nicht aus dem vorhandenen Bucket in den neuen Bucket

Um die Einstellungen eines vorhandenen Buckets zu kopieren, wählen Sie unter Einstellungen aus vorhandenem Bucket kopieren die Option Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geöffnet. Suchen Sie den Bucket mit den Einstellungen, die Sie kopieren möchten, und wählen Sie Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geschlossen, und das Fenster „Bucket erstellen“ wird erneut geöffnet.

Unter Einstellungen aus vorhandenem Bucket kopieren wird nun der Name des ausgewählten Buckets angezeigt. Außerdem wird die Option Standardwerte wiederherstellen angezeigt, mit der Sie die kopierten Bucket-Einstellungen entfernen können. Überprüfen Sie die verbleibenden Bucket-Einstellungen auf der Seite Bucket erstellen. Sie werden sehen, dass sie jetzt mit den Einstellungen des von Ihnen ausgewählten Buckets übereinstimmen. Sie können mit dem letzten Schritt fortfahren.

9. Wählen Sie unter Object Ownership eine der folgenden Einstellungen aus, um die Inhaberschaft von Objekten, die in Ihren Bucket hochgeladen wurden, zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

ACLsdeaktiviert

- Bucket Owner enforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

Standardmäßig ACLs sind sie deaktiviert. Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option weiterhin zu deaktivieren, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.

Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control`ACL, können Sie [eine Bucket-Richtlinie hinzufügen](#), die nur Objekt-Uploads zulässt, die dies verwenden. ACL


- Object Writer — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Note

Die Standardeinstellung ist Bucket-Eigentümer erzwungen. Um die Standardeinstellung anzuwenden und ACLs deaktiviert zu lassen, ist nur die `s3:CreateBucket` entsprechende Berechtigung erforderlich. Um diese Option zu aktivierenACLs, benötigen Sie die `s3:PutBucketOwnershipControls` entsprechende Genehmigung.

10. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.

Alle vier Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

 Note

Zum Aktivieren aller Einstellungen zum Blockieren des öffentlichen Zugriffs ist nur die `s3:CreateBucket`-Berechtigung erforderlich. Wenn Sie eine der Einstellungen zum Blockieren des öffentlichen Zugriffs deaktivieren möchten, benötigen Sie die `s3:PutBucketPublicAccessBlock`-Berechtigung.

11. (Optional) Unter Bucket Versioning (Bucket-Versionsverwaltung) können Sie auswählen, ob Sie Varianten von Objekten in Ihrem Bucket beibehalten möchten. Weitere Informationen über das Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).


Wenn Sie die Versionsverwaltung in Ihrem Bucket deaktivieren oder aktivieren möchten, wählen Sie entweder Disable (Deaktivieren) oder Enable (Aktivieren) aus.

12. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher verwendet werden.

Wenn Sie ein Bucket-Tag hinzuzufügen, geben Sie einen Key (Schlüssel) und optional einen Value (Wert) ein. Wählen Sie dann Add Tag (Tag hinzufügen) aus.

13. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
14. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Standardverschlüsselung zu konfigurieren:

- Verwalteter Amazon S3 S3-Schlüssel (SSE-S3)
- AWS Key Management Service Schlüssel (SSE-KMS)

 Important

Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, unterliegen Sie dem Kontingent für Anfragen pro Sekunde (RPS) von AWS KMS. Für weitere Informationen über AWS

KMS Kontingente und wie Sie eine Quotenerhöhung beantragen können, finden Sie unter [Kontingente](#) im AWS Key Management Service Leitfaden für Entwickler.

Buckets und neue Objekte werden mit serverseitiger Verschlüsselung verschlüsselt. Dabei ist ein Von Amazon S3 verwalteter Schlüssel die Grundebene der Verschlüsselungskonfiguration. Weitere Informationen zur Standardverschlüsselung finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

15. Wenn du wählst AWS Key Management Service Taste (SSE-KMS), gehen Sie wie folgt vor:
 - a. Unter AWS KMS key, geben Sie Ihren KMS Schlüssel auf eine der folgenden Arten an:
 - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie Ihren KMS Schlüssel aus der Liste der verfügbaren Schlüssel aus.
- Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.
- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
 - Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

 **Important**

Sie können nur KMS Schlüssel verwenden, die in derselben Version verfügbar sind AWS-Region wie der Eimer. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS

Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [KMSSchlüssel erstellen, die andere Konten verwenden können](#), in AWS Key Management Service Leitfaden für Entwickler. Weitere Informationen zu SSE - finden KMS Sie unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#).

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter [Identifizieren symmetrischer und KMS asymmetrischer Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler. Weitere Informationen zur Verwendung von AWS KMS mit Amazon S3, siehe [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

- b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren. S3-Bucket Keys senken die Verschlüsselungskosten, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

16. (Optional) Wenn Sie die S3-Objektsperre aktivieren möchten, gehen Sie wie folgt vor:

- a. Wählen Sie Erweiterte Einstellungen aus.

 **Important**

Durch Aktivieren der Objektsperre wird auch die Versioning für den Bucket aktiviert. Nach dem Aktivieren müssen Sie die Standardeinstellungen für die Objektsperre im

Hinblick auf die (rechtliche) Aufbewahrung konfigurieren, um neue Objekte vor dem Löschen oder Überschreiben zu schützen.

- b. Wenn Sie die Objektsperre aktivieren möchten, wählen Sie Enable (Aktivieren) aus, lesen Sie die angezeigte Warnung und bestätigen Sie sie.

Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Note

Wenn Sie einen Bucket mit aktivierter Objektsperre erstellen möchten, benötigen Sie die folgenden Berechtigungen: `s3:CreateBucket`, `s3:PutBucketVersioning` und `s3:PutBucketObjectLockConfiguration`.

17. Wählen Sie Bucket erstellen aus.

Verwendung der AWS SDKs

Wenn Sie das verwenden AWS SDKsUm einen Bucket zu erstellen, müssen Sie einen Client erstellen und dann den Client verwenden, um eine Anfrage zur Erstellung eines Buckets zu senden. Es hat sich bewährt, dass Sie Ihren Client und Ihren Bucket im selben System erstellen AWS-Region. Wenn Sie bei der Erstellung eines Clients oder eines Buckets keine Region angeben, verwendet Amazon S3 die Standardregion USA Ost (Nord-Virginia). Wenn Sie die Bucket-Erstellung auf einen bestimmten Wert beschränken möchten AWS-Region, verwenden Sie den [LocationConstraint](#)Bedingungsschlüssel.

Um einen Client für den Zugriff auf einen Dual-Stack-Endpunkt zu erstellen, müssen Sie einen angeben AWS-Region. Weitere Informationen finden Sie unter[Dual-Stack-Endpunkte](#). Eine Liste der verfügbaren AWS-Regionen, siehe [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

Wenn Sie einen Client erstellen, wird die Region dem regionsspezifischen Endpunkt zugeordnet. Der Client verwendet diesen Endpunkt für die Kommunikation mit Amazon S3: `s3.region.amazonaws.com`. Wenn Ihre Region nach dem 20. März 2019 gestartet wurde, müssen sich Ihr Client und Ihr Bucket in derselben Region befinden. Sie können jedoch einen Client in der Region USA Ost (Nord-Virginia) verwenden, um einen Bucket in einer beliebigen Region zu erstellen, die vor dem 20. März 2019 gestartet wurde. Weitere Informationen finden Sie unter [Legacy-Endpunkte](#).

Diese AWS SDKCodebeispiele erfüllen die folgenden Aufgaben:

- Erstellen Sie einen Client, indem Sie explizit einen angeben AWS-Region— In dem Beispiel verwendet der Client den `s3.us-west-2.amazonaws.com` Endpunkt, um mit Amazon S3 zu kommunizieren. Sie können eine beliebige angeben AWS-Region. Für eine Liste von AWS-Regionen, siehe [Regionen und Endpunkte](#) in der AWS Allgemeine Referenz.
- Senden einer Bucket-Erstellungs-Anfrage durch Angabe eines Bucket-Namens – Der Client sendet eine Anfrage an Amazon S3, um den Bucket in der Region zu erstellen, in der Sie einen Client erstellt haben.
- Abrufen von Informationen zum Standort des Buckets – Amazon S3 speichert Informationen zum Standort des Buckets in der Subressource Standort, die dem Bucket zugeordnet ist.

Java

Dieses Beispiel zeigt, wie Sie einen Amazon S3 S3-Bucket mit dem erstellen AWS SDK for Java. Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.GetBucketLocationRequest;

import java.io.IOException;

public class CreateBucket2 {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
```

```
        .build();

        if (!s3Client.doesBucketExistV2(bucketName)) {
            // Because the CreateBucketRequest object doesn't specify a region,
the           // bucket is created in the region specified in the client.
            s3Client.createBucket(new CreateBucketRequest(bucketName));

            // Verify that the bucket was created by retrieving it and checking
its           // location.
            String bucketLocation = s3Client.getBucketLocation(new
GetBucketLocationRequest(bucketName));
            System.out.println("Bucket location: " + bucketLocation);
        }
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
// it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Informationen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [AWS SDK für .NET API Referenz](#) zur Version 3.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.S3.Util;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
```

```
class CreateBucketTest
{
    private const string bucketName = "*** bucket name ***";
    // Specify your bucket region (an example region is shown).
    private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
    private static IAmazonS3 s3Client;
    public static void Main()
    {
        s3Client = new AmazonS3Client(bucketRegion);
        CreateBucketAsync().Wait();
    }

    static async Task CreateBucketAsync()
    {
        try
        {
            if (!(await AmazonS3Util.DoesS3BucketExistAsync(s3Client,
bucketName)))
            {
                var putBucketRequest = new PutBucketRequest
                {
                    BucketName = bucketName,
                    UseClientRegion = true
                };

                PutBucketResponse putBucketResponse = await
s3Client.PutBucketAsync(putBucketRequest);
            }
            // Retrieve the bucket location.
            string bucketLocation = await FindBucketLocationAsync(s3Client);
        }
        catch (AmazonS3Exception e)
        {
            Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
    }

    static async Task<string> FindBucketLocationAsync(IAmazonS3 client)
```

```
    {
        string bucketLocation;
        var request = new GetBucketLocationRequest()
        {
            BucketName = bucketName
        };
        GetBucketLocationResponse response = await
client.GetBucketLocationAsync(request);
        bucketLocation = response.Location.ToString();
        return bucketLocation;
    }
}
```

Ruby

Informationen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [AWS SDK für Ruby — Version 3](#).

Example

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name.
  # This is a client-side object until
  # create is called.
  def initialize(bucket)
    @bucket = bucket
  end

  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)
    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
    false
  end
end
```



```
end

# Gets the Region where the bucket is located.
#
# @return [String] The location of the bucket.
def location
  if @bucket.nil?
    "None. You must create a bucket before you can get its location!"
  else
    @bucket.client.get_bucket_location(bucket: @bucket.name).location_constraint
  end
rescue Aws::Errors::ServiceError => e
  "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
end
end

# Example usage:
def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("amzn-s3-demo-bucket-
#{Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{wrapper.bucket.name}."
  puts "Your bucket's region is: #{wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__
```

Verwendung der AWS CLI

Sie können auch die verwenden AWS Command Line Interface (AWS CLI), um einen S3-Bucket zu erstellen. Weitere Informationen finden Sie unter [create-bucket](#) im AWS CLI Befehlsreferenz.

Für Informationen über AWS CLI, siehe [Was ist der AWS Command Line Interface?](#) in der AWS Command Line Interface Benutzerleitfaden.

Anzeigen der Eigenschaften eines S3-Buckets

Sie können Eigenschaften für jeden Amazon S3 S3-Bucket anzeigen, den Sie besitzen. Diese Einstellungen umfassen u. a. folgende:

- **Bucket Versioning (Bucket-Versionssteuerung)** – Behalten Sie mehrere Versionen eines Objekts in einem Bucket bei, indem Sie die Versionssteuerung verwenden. Für einen neuen Bucket ist das Versioning standardmäßig deaktiviert. Weitere Informationen über die Aktivierung des Versionings finden Sie unter [Aktivieren des Versioning für Buckets](#).
- **Schlagworte** — Mit AWS Bei der Kostenzuweisung können Sie Bucket-Tags verwenden, um die Abrechnung für Ihre Nutzung eines Buckets mit Anmerkungen zu versehen. Ein Tag ist ein Schlüssel-Wert-Paar, das eine Bezeichnung repräsentiert, die Sie einem Bucket zuweisen. Weitere Informationen finden Sie unter [Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets](#).
- **Default encryption (Standardverschlüsselung)** – Die Aktivierung der Standardverschlüsselung bietet Ihnen eine automatische serverseitige Verschlüsselung. Amazon S3 verschlüsselt ein Objekt, bevor es auf einer Festplatte gespeichert wird, und entschlüsselt das Objekt beim Herunterladen. Weitere Informationen finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).
- **Server access logging (Server-Zugriffsprotokollierung)** – Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an Ihren Bucket gestellt wurden. Standardmäßig erfasst Amazon S3 keine Server-Zugriffsprotokolle. Weitere Informationen zur Aktivierung der Server-Zugriffsprotokollierung finden Sie unter [Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung](#).
- **AWS CloudTrail Datenereignisse** — Wird verwendet CloudTrail , um Datenereignisse zu protokollieren. Standardmäßig werden Datenereignisse nicht von den Trails protokolliert. Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen für Trails](#) im AWS CloudTrail Benutzerleitfaden.
- **Event notifications (Ereignisbenachrichtigungen)Ereignisse** – Sie können bestimmte Amazon S3-Bucket-Ereignisse aktivieren, um eine Benachrichtigungsmeldung an ein Ziel zu senden, wenn das Ereignis auftritt. Weitere Informationen finden Sie unter [Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole](#).
- **Transfer Acceleration** – Ermöglicht schnelle, einfache und sichere Übertragungen von Dateien zwischen Ihrem Client und einem S3-Bucket über große Entfernungen. Weitere Informationen zur Aktivierung der Transfer Acceleration finden Sie unter [Aktivieren und Verwenden von S3 Transfer Acceleration](#).
- **Object Lock** – Mit der S3-Objektsperre können Sie für einen festen Zeitraum oder auf unbegrenzte Zeit verhindern, dass ein Objekt gelöscht oder überschrieben wird. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).
- **Requester Pays (Zahlung durch den Anforderer)** – Sie können die Zahlung durch den Anforderer aktivieren, sodass der Anforderer (anstelle des Bucket-Eigentümers) für die Anforderungen und

Datenübertragungen zahlt. Weitere Informationen finden Sie unter [Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung](#).

- Static website hosting (Hosting statischer Websites) – Sie können in Amazon S3 eine statische Website hosten. Weitere Informationen finden Sie unter [Hosten einer statischen Website mit Amazon S3](#).

Sie können die Eigenschaften von Buckets mit dem Befehl anzeigen AWS Management Console, AWS CLI, oder AWS SDKs

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets) den Namen des Buckets aus, dessen Eigenschaften Sie anzeigen wollen.
3. Wählen Sie die Registerkarte Eigenschaften aus.
4. Auf der Eigenschaftenseite können Sie die oben genannten Eigenschaften für den Bucket konfigurieren.

Verwendung der AWS CLI

Zeigen Sie Bucket-Eigenschaften mit dem AWS CLI

Die folgenden Befehle zeigen, wie Sie das verwenden können AWS CLI um verschiedene Bucket-Eigenschaften aufzulisten.

Im Folgenden wird der Tagsatz zurückgegeben, der dem Bucket zugeordnet ist *amzn-s3-demo-bucket1*. Weitere Informationen zu Bucket-Tags finden Sie unter [Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets](#).

```
aws s3api get-bucket-tagging --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter [get-bucket-tagging](#) in der AWS CLI Befehlsreferenz.

Im Folgenden wird der Versionsstatus des Buckets zurückgegeben *amzn-s3-demo-bucket1*. Hinweise zur Bucket-Versionierung finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

```
aws s3api get-bucket-versioning --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter [get-bucket-versioning](#) in der AWS CLI Befehlsreferenz.

Im Folgenden wird die Standardverschlüsselungskonfiguration für den Bucket zurückgegeben *amzn-s3-demo-bucket1*. Standardmäßig haben alle Buckets eine Standardverschlüsselungskonfiguration, die serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwendet. Informationen zur Standardverschlüsselung des Buckets finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#)

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter [get-bucket-encryption](#) in der AWS CLI Befehlsreferenz.

Im Folgenden wird die Benachrichtigungskonfiguration des Buckets zurückgegeben *amzn-s3-demo-bucket1*. Informationen zu den Bucket-Ereignisbenachrichtigungen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

```
aws s3api get-bucket-notification-configuration --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter [get-bucket-notification-configuration](#) in der AWS CLI Befehlsreferenz.

Im Folgenden wird der Logging-Status für den Bucket zurückgegeben *amzn-s3-demo-bucket1*. Hinweise zur Bucket-Protokollierung finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

```
aws s3api get-bucket-logging --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter [get-bucket-logging](#) in der AWS CLI Befehlsreferenz.

Verwendung der AWS SDKs

Beispiele für die Rückgabe von Bucket-Eigenschaften mit dem AWS SDKs, wie Versionierung, Tags und mehr, finden Sie unter [Aktionen für Amazon S3 mit AWS SDKs](#) .

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Leeren eines Buckets

Sie können den Inhalt eines Buckets mit der Amazon S3 S3-Konsole leeren, AWS SDKs, oder AWS Command Line Interface (AWS CLI). Wenn Sie einen Bucket leeren, löschen Sie alle Objekte, behalten aber den Bucket. Nachdem Sie einen Bucket geleert haben, kann es nicht rückgängig gemacht werden. Dem Bucket während der Löschaktion hinzugefügte Objekte werden möglicherweise gelöscht. Alle Objekte (einschließlich aller Objektversionen und Löschmarkierungen) im Bucket müssen gelöscht werden, bevor der Bucket selbst gelöscht werden kann.

Wenn Sie einen Bucket mit aktivierter oder ausgesetzter S3-Versionsverwaltung leeren, werden alle Versionen aller Objekte im Bucket gelöscht. Weitere Informationen finden Sie unter [Arbeiten mit Objekten in einem versioning-fähigen Bucket](#).

Sie können auch eine Lebenszyklus-Konfiguration für einen Bucket angeben, sodass Objekte ablaufen, damit Amazon S3 sie löschen kann. Weitere Informationen finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#). Um einen großen Bucket zu leeren, empfehlen wir, eine S3-Lifecycle-Konfigurationsregel zu verwenden. Der Ablauf des Lebenszyklus ist ein asynchroner Prozess, daher kann es einige Tage dauern, bis die Regel ausgeführt wird und der Bucket leer ist. Nachdem Amazon S3 die Regel zum ersten Mal ausführt, werden alle Objekte, die für den Ablauf in Frage kommen, zum Löschen markiert. Die Objekte, die zum Löschen markiert sind, werden Ihnen nicht mehr in Rechnung gestellt. Weitere Informationen finden Sie unter [Wie leere ich einen Amazon-S3-Bucket mithilfe einer Lebenszykluskonfigurationsregel?](#).

Verwenden der S3-Konsole

Sie können die Amazon-S3-Konsole verwenden, um einen Bucket zu leeren, der alle Objekte im Bucket löscht, ohne den Bucket zu löschen.

Einen S3-Bucket leeren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) die Option neben dem Namen des Buckets aus, den Sie leeren möchten. Wählen Sie anschließend Empty (Leeren) aus.
3. Bestätigen Sie auf der Seite Empty bucket (Leerer Bucket), dass Sie den Bucket leeren möchten, indem Sie den Bucket-Namen in das Textfeld eingeben, und wählen Sie dann Empty (Leeren).

- Überwachen Sie den Fortschritt des Vorgangs zum Entleeren von Buckets auf der Seite `Empty bucket: Status` (`Bucket entleeren: Status`).

Verwendung der AWS CLI

Sie können einen Bucket leeren, indem Sie AWS CLI nur wenn für den Bucket die Bucket-Versionierung nicht aktiviert ist. Wenn die Versionierung nicht aktiviert ist, können Sie die Option `rm` (entfernen) verwenden AWS CLI Befehl mit dem `--recursive` Parameter zum Leeren des Buckets (oder zum Entfernen einer Teilmenge von Objekten mit einem bestimmten Schlüsselnamenpräfix).

Der folgende `rm`-Befehl entfernt Objekte mit dem Schlüsselnamenpräfix `doc`, z. B. `doc/doc1` und `doc/doc2`.

```
$ aws s3 rm s3://bucket-name/doc --recursive
```

Verwenden Sie den folgenden Befehl, um alle Objekte zu entfernen, ohne ein Präfix anzugeben.

```
$ aws s3 rm s3://bucket-name --recursive
```

Weitere Informationen finden Sie unter [Verwenden von S3-Befehlen auf hoher Ebene mit dem AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Note

Sie können keine Objekte aus einem Bucket entfernen, für den die Versionierung aktiviert ist. Amazon S3 fügt eine Löschmarkierung hinzu, wenn Sie ein Objekt löschen, was dieser Befehl tut. Weitere Informationen über die S3-Bucket-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Verwendung der AWS SDKs

Sie können das AWS SDKsum einen Bucket zu leeren oder eine Teilmenge von Objekten zu entfernen, die ein bestimmtes Schlüsselnamenpräfix haben.

Ein Beispiel für das Leeren eines Buckets finden Sie mit AWS SDK for Java, finden Sie unter [Löschen eines Buckets](#). Der Code löscht alle Objekte, unabhängig davon, ob für den Bucket das

Versioning aktiviert ist. Anschließend löscht er den Bucket. Um den Bucket nur zu leeren, stellen Sie sicher, dass Sie die Anweisung entfernen, die den Bucket löscht.

Weitere Informationen zur Verwendung von anderen AWS SDKs, siehe [Tools für Amazon Web Services](#).

Verwenden einer Lebenszyklus-Konfiguration

Um einen großen Bucket zu leeren, empfehlen wir, eine S3-Lifecycle-Konfigurationsregel zu verwenden. Der Ablauf des Lebenszyklus ist ein asynchroner Prozess, daher kann es einige Tage dauern, bis die Regel ausgeführt wird und der Bucket leer ist. Nachdem Amazon S3 die Regel zum ersten Mal ausführt, werden alle Objekte, die für den Ablauf in Frage kommen, zum Löschen markiert. Die Objekte, die zum Löschen markiert sind, werden Ihnen nicht mehr in Rechnung gestellt. Weitere Informationen finden Sie unter [Wie leere ich einen Amazon-S3-Bucket mithilfe einer Lebenszykluskonfigurationsregel?](#).

Wenn Sie eine Lebenszykluskonfiguration verwenden, um Ihren Bucket zu leeren, sollte die Konfiguration [aktuelle Versionen](#), [nicht aktuelle Versionen](#), [Löschmarkierungen](#) und [unvollständige mehrteilige Uploads](#) enthalten.

Sie können Lebenszyklus-Konfigurationsregeln hinzufügen, sodass alle Objekte oder eine Untermenge davon mit einem spezifischen Schlüsselnamenpräfix ablaufen. Sie können z. B. eine Lebenszyklusregel so festlegen, dass Objekte einen Tag nach dem Erstellungsdatum ablaufen, um alle Objekte in einem Bucket zu entfernen.

Amazon S3 unterstützt eine Bucket-Lebenszyklusregel, mit der Sie unvollständige mehrteilige Uploads abrechnen können, die nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wir empfehlen, dass Sie diese Lebenszyklusregel konfigurieren, um Ihre Speicherkosten zu minimieren. Weitere Informationen finden Sie unter [Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads](#).

Weitere Informationen zur Verwendung einer Lebenszyklus-Konfiguration zum Leeren eines Buckets finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#) und [Auslaufende Objekte](#).

Leeren eines Buckets mit AWS CloudTrail konfiguriert

AWS CloudTrail verfolgt Datenereignisse auf Objektebene in einem Amazon S3 S3-Bucket, z. B. das Löschen von Objekten. Wenn Sie einen Bucket als Ziel für die Protokollierung Ihrer CloudTrail Ereignisse verwenden und Objekte aus demselben Bucket löschen, erstellen Sie möglicherweise

neue Objekte, während Sie Ihren Bucket leeren. Um dies zu verhindern, beenden Sie Ihre AWS CloudTrail Pfade. Weitere Informationen darüber, wie du verhindern kannst, dass deine CloudTrail Trails Ereignisse protokollieren, findest du unter [Deaktivierung der Protokollierung für einen Trail](#) in AWS CloudTrail Benutzerleitfaden.

Eine weitere Alternative, um zu verhindern, dass CloudTrail Trails zum Bucket hinzugefügt werden, besteht darin, Ihrer Bucket-Richtlinie eine `s3:PutObject Deny`-Anweisung hinzuzufügen. Wenn Sie zu einem späteren Zeitpunkt neue Objekte im Bucket speichern möchten, müssen Sie diese `Deny s3:PutObject`-Anweisung entfernen. Weitere Informationen finden Sie unter [Objektoperationen](#) und zu den [IAMJSONRichtlinienelementen: Wirkung](#) im IAMBenutzerhandbuch.

Löschen eines Buckets

Sie können einen leeren Amazon-S3-Bucket löschen. Berücksichtigen Sie Folgendes, bevor Sie einen Bucket löschen:

- Bucket-Namen sind eindeutig. Wenn Sie einen Bucket löschen, wird ein anderer AWS Benutzer kann den Namen verwenden.
- Wenn der Bucket eine statische Website hostet und Sie eine gehostete Zone von Amazon Route 53, wie unter [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#) beschrieben, erstellt und konfiguriert haben, müssen Sie die gehosteten Route 53-Zoneneinstellungen bereinigen, die sich auf den Bucket beziehen. Weitere Informationen finden Sie unter [Schritt 2: Löschen der von Route 53 gehosteten Zone](#).
- Wenn der Bucket Logdaten von Elastic Load Balancing (ELB) empfängt: Wir empfehlen, dass Sie die Übermittlung von ELB Logs an den Bucket beenden, bevor Sie ihn löschen. Erstellt ein anderer Benutzer nach dem Löschen des Buckets einen Bucket mit demselben Namen, dann könnten Ihre Protokolldaten potenziell an diesen Bucket übermittelt werden. Informationen zu ELB Zugriffs-Logs finden Sie unter [Access Logs](#) im User Guide for Classic Load Balancers und [Access Logs](#) im User Guide for Application Load Balancers.

Fehlerbehebung

Wenn Sie einen Amazon-S3-Bucket nicht löschen können, sollten Sie Folgendes berücksichtigen:

- Stellen Sie sicher, dass der Bucket leer ist – Sie können nur Buckets löschen, bei denen keine Objekte enthalten sind. Stellen Sie sicher, dass der Bucket leer ist.

- Stellen Sie sicher, dass keine Zugriffspunkte angefügt sind – Sie können nur Buckets löschen, denen keine Zugriffspunkte angefügt sind. Löschen Sie alle Zugriffspunkte, die dem Bucket angefügt sind, bevor Sie den Bucket löschen.
- AWS Organizations Dienststeuerungsrichtlinien (SCPs) — Eine Dienststeuerungsrichtlinie kann die Löschberechtigung für einen Bucket verweigern. Weitere Informationen dazu SCPs finden Sie unter [Servicesteuerungsrichtlinien](#) im AWS Organizations Benutzerleitfaden.
- s3: DeleteBucket Berechtigungen — Wenn Sie einen Bucket nicht löschen können, überprüfen Sie gemeinsam mit Ihrem IAM Administrator, ob Sie über die `s3:DeleteBucket` erforderlichen Berechtigungen verfügen. Informationen zum Anzeigen oder Aktualisieren von IAM Berechtigungen finden Sie im [IAMBenutzerhandbuch unter Ändern von Berechtigungen für einen IAM Benutzer](#).
- s3: Anweisung DeleteBucket verweigern — Wenn Ihre IAM Richtlinie über `s3:DeleteBucket` Berechtigungen verfügt und Sie einen Bucket nicht löschen können, enthält die Bucket-Richtlinie möglicherweise eine Ablehnungsaussage für `s3:DeleteBucket`. Buckets, die von erstellt wurden ElasticBeanstalk , verfügen standardmäßig über eine Richtlinie, die diese Aussage enthält. Bevor Sie den Bucket löschen können, müssen Sie diese Anweisung oder die Bucket-Richtlinie löschen.

Important

Bucket-Namen sind eindeutig. Wenn Sie einen Bucket löschen, wird ein anderer AWS Benutzer den Namen verwenden. Wenn Sie weiterhin denselben Bucket-Namen verwenden wollen, sollten Sie den Bucket nicht löschen. Wir empfehlen, den Bucket zu leeren und beizubehalten.

Verwenden der S3-Konsole

Einen S3-Bucket löschen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets die Option neben dem Namen des Buckets aus, den Sie löschen möchten, und wählen Sie dann oben auf der Seite Delete (Löschen) aus.
3. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Note

Wenn der Bucket Objekte enthält, leeren Sie den Bucket, bevor Sie ihn löschen, indem Sie in der Fehlermeldung This bucket is not empty (Dieser Bucket ist nicht leer) den Link für eine leere Bucket-Konfiguration auswählen und die Anweisungen auf der Seite Empty bucket (Bucket leeren) befolgen. Kehren Sie dann zur Seite Delete bucket (Bucket löschen) zurück und löschen Sie den Bucket.

4. Wenn Sie überprüfen möchten, ob Sie den Bucket gelöscht haben, öffnen Sie die Buckets-Liste und geben den Namen des Buckets ein, den Sie gelöscht haben. Wenn der Buckets nicht gefunden wird, war der Löschvorgang erfolgreich.

Verwendung der AWS SDK für Java

Das folgende Beispiel zeigt Ihnen, wie Sie einen Bucket mit dem löschen AWS SDK für Java. Zuerst löscht der Code Objekte im Bucket, und dann löscht er das Bucket. Für Informationen über andere AWS SDKs, siehe [Tools für Amazon Web Services](#).

Java

Das folgende Java-Beispiel löscht einen Bucket mit Objekten. Zuerst löscht das Beispiel alle Objekte und dann den Bucket. Das Beispiel funktioniert für Buckets mit oder ohne aktiviertem Versioning.

Note

Bei Buckets ohne aktiviertes Versioning können Sie alle Objekte direkt löschen und danach den Bucket löschen. Bei Buckets mit aktiviertem Versioning müssen Sie zuerst alle Objektversionen löschen, bevor Sie den Bucket löschen.

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;
```

```
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.Iterator;

public class DeleteBucket2 {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Delete all objects from the bucket. This is sufficient
            // for unversioned buckets. For versioned buckets, when you attempt to
delete
            // objects, Amazon S3 inserts
            // delete markers for all objects, but doesn't delete the object
versions.
            // To delete objects from versioned buckets, delete all of the object
versions
            // before deleting
            // the bucket (see below for an example).
            ObjectListing objectListing = s3Client.listObjects(bucketName);
            while (true) {
                Iterator<S3ObjectSummary> objIter =
objectListing.getObjectSummaries().iterator();
                while (objIter.hasNext()) {
                    s3Client.deleteObject(bucketName, objIter.next().getKey());
                }

                // If the bucket contains many objects, the listObjects() call
                // might not return all of the objects in the first listing. Check
to
                // see whether the listing was truncated. If so, retrieve the next
page of
                // objects
            }
        }
    }
}
```

```
        // and delete them.
        if (objectListing.isTruncated()) {
            objectListing = s3Client.listNextBatchOfObjects(objectListing);
        } else {
            break;
        }
    }

    // Delete all object versions (required for versioned buckets).
    VersionListing versionList = s3Client.listVersions(new
ListVersionsRequest().withBucketName(bucketName));
    while (true) {
        Iterator<S3VersionSummary> versionIter =
versionList.getVersionSummaries().iterator();
        while (versionIter.hasNext()) {
            S3VersionSummary vs = versionIter.next();
            s3Client.deleteVersion(bucketName, vs.getKey(),
vs.getVersionId());
        }

        if (versionList.isTruncated()) {
            versionList = s3Client.listNextBatchOfVersions(versionList);
        } else {
            break;
        }
    }

    // After all objects and object versions are deleted, delete the bucket.
    s3Client.deleteBucket(bucketName);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
couldn't
    // parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Verwendung der AWS CLI

Sie können einen Bucket löschen, der Objekte enthält, mit AWS CLI wenn die Versionierung nicht aktiviert ist. Wenn Sie einen Bucket löschen, der Objekte enthält, werden alle Objekte in dem Bucket dauerhaft gelöscht. Dazu gehören auch Objekte, die in die S3 Glacier-Speicherklasse übergegangen sind.

Wenn in Ihrem Bucket die Versionierung nicht aktiviert ist, können Sie `rb` (Bucket entfernen) verwenden AWS CLI Befehl mit dem `--force` Parameter zum Löschen des Buckets und aller darin enthaltenen Objekte. Dieser Befehl löscht zuerst alle Objekte und dann den Bucket.

Wenn das Versioning aktiviert ist, werden versionierte Objekte in diesem Prozess nicht gelöscht. Dadurch würde das Löschen des Buckets fehlschlagen, da der Bucket nicht leer wäre. Informationen zum Löschen versionierter Objekte finden Sie unter [Löschen von Objektversionen](#).

```
$ aws s3 rb s3://bucket-name --force
```

Weitere Informationen finden Sie unter [Verwenden von S3-Befehlen auf hoher Ebene mit AWS Command Line Interface](#) in der AWS Command Line Interface Benutzerleitfaden.

Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets


Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Für alle Amazon S3 S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert, und Objekte werden automatisch mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-

Schlüsseln (SSE-S3) verschlüsselt. Diese Verschlüsselungseinstellung gilt für alle Objekte in Ihren Amazon-S3-Buckets.

Wenn Sie mehr Kontrolle über Ihre Schlüssel benötigen, z. B. bei der Verwaltung der Schlüsselrotation und der Gewährung von Zugriffsrichtlinien, können Sie sich für die serverseitige Verschlüsselung mit entscheiden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS). Weitere Informationen zum Bearbeiten von KMS Schlüsseln finden Sie unter [Bearbeiten von Schlüsseln](#) in AWS Key Management Service Entwicklerhandbuch.

 Note

Wir haben die Buckets geändert, sodass neue Objekt-Uploads automatisch verschlüsselt werden. Wenn Sie zuvor einen Bucket ohne Standardverschlüsselung erstellt haben, aktiviert Amazon S3 standardmäßig die Verschlüsselung für den Bucket mit SSE -S3. Für einen vorhandenen Bucket, für den bereits SSE -S3 oder SSE - KMS konfiguriert ist, werden keine Änderungen an der Standardverschlüsselungskonfiguration vorgenommen. Wenn Sie Ihre Objekte mit SSE - verschlüsseln möchtenKMS, müssen Sie den Verschlüsselungstyp in Ihren Bucket-Einstellungen ändern. Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren, um den Anforderungsverkehr von Amazon S3 zu reduzieren AWS KMS und reduzieren Sie die Kosten für die Verschlüsselung. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Um Buckets zu identifizieren, für die SSE die Standardverschlüsselung KMS aktiviert ist, können Sie Amazon S3 Storage Lens-Metriken verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter [Verwenden von S3 Storage Lens zum Schutz Ihrer Daten](#).

Wenn Sie die serverseitige Verschlüsselung verwenden, verschlüsselt Amazon S3 ein Objekt vor dem Speichern auf der Festplatte und entschlüsselt es beim Herunterladen des Objekts. Weitere Informationen zum Schutz von Daten mithilfe der serverseitigen Verschlüsselung und der Verwaltung der Verschlüsselungsschlüssel finden Sie unter [Schützen von Daten mit serverseitiger Verschlüsselung](#).

Weitere Informationen zu den für die Standardverschlüsselung erforderlichen Berechtigungen finden Sie unter [PutBucketEncryption](#) in der Amazon Simple Storage Service API Reference.

Sie können das standardmäßige Verschlüsselungsverhalten von Amazon S3 für einen S3-Bucket konfigurieren, indem Sie die Amazon S3 S3-Konsole verwenden, die AWS SDKs, der Amazon S3 REST API und der AWS Befehlszeilenschnittstelle (AWS CLI).

Verschlüsseln vorhandener Objekte

Zum Verschlüsseln Ihrer vorhandenen nicht verschlüsselten Amazon-S3-Objekte können Sie Amazon S3 Batch Operations verwenden. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, mit denen gearbeitet werden soll, und Batch Operations ruft die entsprechenden Objekte auf API, um die angegebene Operation auszuführen. Mit der Operation [Batch Operations Copy](#) können Sie vorhandenen nicht verschlüsselte Objekte kopieren und die neuen verschlüsselten Objekte in denselben Bucket schreiben. Ein einzelner Batchoperations-Auftrag kann die angegebene Operation für Milliarden von Objekten ausführen. Weitere Informationen finden Sie unter [Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch](#) und im AWS Speicher-Blogbeitrag [Verschlüsseln von Objekten mit Amazon S3 Batch Operations](#).

Sie können vorhandene Objekte auch verschlüsseln, indem Sie den CopyObject API Vorgang oder den copy-object AWS CLI Befehl. Weitere Informationen finden Sie hier: AWS Speicher-Blogbeitrag [Verschlüsseln vorhandener Amazon S3 S3-Objekte mit dem AWS CLI](#).

Note

Amazon S3 S3-Buckets, bei denen die Standard-Bucket-Verschlüsselung auf SSE — eingestellt ist, KMS können nicht als Ziel-Buckets für verwendet werden. [the section called "Protokollierungs-Serverzugriff"](#) Für SSE Ziel-Buckets des Serverzugriffsprotokolls wird nur die Standardverschlüsselung -S3 unterstützt.

Verwendung von SSE — KMS Verschlüsselung für kontoübergreifende Operationen

Beachten Sie Folgendes, wenn Sie kontoübergreifende Operationen verschlüsseln:

- Wenn ein AWS KMS key Der Amazon-Ressourcenname (ARN) oder Alias wird bei der Anfrage oder über die Standardverschlüsselungskonfiguration des Buckets, die Von AWS verwalteter Schlüssel (aws/s3) wird verwendet.

- Wenn Sie S3-Objekte hochladen oder auf sie zugreifen, indem Sie AWS Identity and Access Management (IAM) Prinzipale, die sich im selben befinden AWS-Konto als KMS Schlüssel können Sie den verwenden Von AWS verwalteter Schlüssel (`aws/s3`).
- Verwenden Sie einen vom Kunden verwalteten Schlüssel, wenn Sie kontoübergreifenden Zugriff auf Ihre S3-Objekte gewähren möchten. Sie können die Richtlinie eines vom Kunden verwalteten Schlüssel so konfigurieren, dass der Zugriff von einem anderen Konto aus möglich ist.
- Wenn Sie einen vom Kunden verwalteten KMS Schlüssel angeben, empfehlen wir die Verwendung eines vollqualifizierten KMS SchlüsselsARN. Wenn Sie stattdessen einen KMS Schlüsselalias verwenden, AWS KMS löst den Schlüssel im Konto des Anfragenden auf. Dieses Verhalten kann dazu führen, dass Daten mit einem KMS Schlüssel verschlüsselt werden, der dem Anforderer und nicht dem Bucket-Besitzer gehört.
- Sie müssen einen Schlüssel angeben, für den Ihnen (dem Anforderer) die Berechtigung Encrypt erteilt wurde. Weitere Informationen finden Sie unter [Erlauben Sie Schlüsselbenutzern, einen KMS Schlüssel für kryptografische Operationen zu verwenden](#) in AWS Key Management Service Leitfaden für Entwickler.

Weitere Informationen darüber, wann Sie vom Kunden verwaltete Schlüssel verwenden sollten, und AWS verwaltete KMS Schlüssel finden Sie unter [Sollte ich einen verwenden Von AWS verwalteter Schlüssel oder ein vom Kunden verwalteter Schlüssel zur Verschlüsselung meiner Objekte in Amazon S3?](#)

Verwenden der Standard-Verschlüsselung mit der Replikation

Wenn Sie die Standard-Verschlüsselung für einen Replikations-Ziel-Bucket aktivieren, gilt das folgende Verschlüsselungsverhalten:

- Wenn Objekte im Quell-Bucket nicht verschlüsselt sind, werden die Replikatobjekte im Ziel-Bucket mithilfe der Einstellungen der Standard-Verschlüsselung des Ziel-Buckets verschlüsselt. Infolgedessen unterscheiden sich die Entity-Tags (ETags) der Quellobjekte ETags von denen der Replikatobjekte. Wenn Sie Anwendungen haben, die dies verwenden ETags, müssen Sie diese Anwendungen aktualisieren, um diesem Unterschied Rechnung zu tragen.
- Wenn Objekte im Quell-Bucket mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt werden, serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS), die Replikatobjekte im Ziel-Bucket

verwenden denselben Verschlüsselungstyp wie die Quellobjekte. Die Einstellungen der Standard-Verschlüsselung des Ziel-Buckets werden nicht verwendet.

Weitere Hinweise zur Verwendung der Standardverschlüsselung mit SSE - finden Sie KMS unter [Replizieren verschlüsselter Objekte](#).

Verwenden von Amazon S3-Bucket-Schlüsseln mit Standard-Verschlüsselung

Wenn Sie Ihren Bucket so konfigurieren, dass er SSE - KMS als Standardverschlüsselungsverhalten für neue Objekte verwendet, können Sie auch S3-Bucket-Keys konfigurieren. S3 Bucket Keys reduzieren die Anzahl der Transaktionen von Amazon S3 auf AWS KMS um die Kosten von SSE - zu senkenKMS.

Wenn Sie Ihren Bucket so konfigurieren, dass er S3 Bucket Keys for SSE - KMS für neue Objekte verwendet, AWS KMS generiert einen Schlüssel auf Bucket-Ebene, der verwendet wird, um einen eindeutigen [Datenschlüssel](#) für Objekte im Bucket zu erstellen. Dieser S3-Bucket Key wird für einen begrenzten Zeitraum innerhalb von Amazon S3 verwendet, sodass Amazon S3 weniger Anfragen stellen muss AWS KMS um Verschlüsselungsvorgänge abzuschließen.

Weitere Informationen zur Verwendung von S3-Bucket-Schlüsseln finden Sie unter [Verwenden von Amazon-S3-Bucket-Schlüssel](#).

Konfigurieren der Standardverschlüsselung

Important


Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

In Amazon S3 S3-Buckets ist die Bucket-Verschlüsselung standardmäßig aktiviert, und neue Objekte werden automatisch mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Diese Verschlüsselung gilt für alle neuen Objekte in Ihren Amazon-S3-Buckets und es fallen keine Kosten für Sie an.

Wenn Sie mehr Kontrolle über Ihre Verschlüsselungsschlüssel benötigen, z. B. die Verwaltung der Schlüsselrotation und die Gewährung von Zugriffsrichtlinien, können Sie sich für die serverseitige Verschlüsselung mit entscheiden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS). Weitere Hinweise zu SSE - finden KMS Sie unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#). Weitere Informationen zu DSSE - finden KMS Sie unter [the section called "Zweischichtige serverseitige Verschlüsselung \(DSSE-\) KMS"](#).

Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, müssen Sie über die entsprechende Berechtigung verfügen. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [Erstellen von KMS Schlüsseln, die andere Konten verwenden können](#), in AWS Key Management Service Leitfaden für Entwickler.

Wenn Sie die Standard-Bucket-Verschlüsselung auf SSE - setzenKMS, können Sie auch einen S3-Bucket-Key konfigurieren, um Ihre AWS KMS Kosten anfordern. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

 Note

Wenn Sie Ihre Standard-Bucket-Verschlüsselung auf SSE - setzenKMS, sollten Sie überprüfen, ob Ihre KMS Schlüssel-ID korrekt ist. [PutBucketEncryption](#) Amazon S3 validiert die in PutBucketEncryption Anfragen angegebene KMS Schlüssel-ID nicht.

Es gibt keine zusätzlichen Gebühren für die Nutzung von Standard-Verschlüsselung für S3-Buckets. Für Anforderungen zum Konfigurieren des Standardverschlüsselungsverhaltens werden Standardgebühren für Amazon-S3-Anforderungen berechnet. Informationen zu Preisen finden Sie unter [Amazon S3 – Preise](#). Für SSE - KMS und DSSE -KMS, AWS KMS Es fallen Gebühren an, die unter aufgeführt sind [AWS KMS Preisgestaltung](#).

Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) wird für die Standardverschlüsselung nicht unterstützt.

Sie können die Amazon S3 S3-Standardverschlüsselung für einen S3-Bucket mithilfe der Amazon S3 S3-Konsole konfigurieren, AWS SDKs, der Amazon S3 REST API und der AWS Command Line Interface (AWS CLI).

Änderungen, die Sie vor dem Aktivieren der Standardverschlüsselung beachten sollten

Nachdem Sie die Standard-Verschlüsselung für einen Bucket aktiviert haben, gilt das folgende Verschlüsselungsverhalten:

- Es gibt keine Änderung der Verschlüsselung der Objekte, die vor der Aktivierung der Standard-Verschlüsselung im Bucket vorhanden waren.
- Wenn Sie Objekte nach der Aktivierung der Standard-Verschlüsselung hochladen:
 - Wenn Ihre PUT-Abfrage-Header keine Verschlüsselungsinformationen mit einschließen, verwendet Amazon S3 die Standardverschlüsselungseinstellungen des Buckets, um die Objekte zu verschlüsseln.
 - Wenn Ihre PUT-Anfrage-Header Verschlüsselungsinformationen mit einschließen, verwendet Amazon S3 die Verschlüsselungsinformationen der PUT-Anfrage, um Objekte zu verschlüsseln, bevor sie in Amazon S3 gespeichert werden.
- Wenn Sie die KMS Option SSE - KMS oder DSSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, gelten für Sie die Kontingente für Anfragen pro Sekunde (RPS) von AWS KMS. Für weitere Informationen über AWS KMS Kontingente und wie Sie eine Quotenerhöhung beantragen können, finden Sie unter [Kontingente](#) im AWS Key Management Service Leitfaden für Entwickler.

Note


Objekte, die hochgeladen wurden, bevor die Standardverschlüsselung aktiviert wurde, werden nicht verschlüsselt. Weitere Informationen zum Verschlüsseln vorhandener Objekte finden Sie unter [the section called “Festlegen der Standard-Bucket-Verschlüsselung”](#).

Verwenden der S3-Konsole

Konfigurieren der Standardverschlüsselung für einen Amazon-S3-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
6. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Verschlüsselung zu konfigurieren:
 - Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)
 - Serverseitige Verschlüsselung mit AWS Key Management Service Schlüssel (SSE-KMS)
 - Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service Schlüssel (-) DSSE KMS

 **Important**

Wenn Sie die KMS Optionen SSE - KMS oder DSSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, gelten für Sie die Kontingente für Anfragen pro Sekunde (RPS) von AWS KMS. Für weitere Informationen über AWS KMS Kontingente und wie Sie eine Quotenerhöhung beantragen können, finden Sie unter [Kontingente](#) im AWS Key Management Service Leitfaden für Entwickler.

Buckets und neue Objekte werden standardmäßig mit SSE -S3 verschlüsselt, sofern Sie keine andere Standardverschlüsselung für Ihre Buckets angeben. Weitere Informationen zur Standardverschlüsselung finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

7. Wenn Sie serverseitige Verschlüsselung mit gewählt haben AWS Key Management Service Schlüssel (SSE-KMS) oder zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service Schlüssel (DSSE-KMS), gehen Sie wie folgt vor:
 - a. Unter AWS KMS key, geben Sie Ihren KMS Schlüssel auf eine der folgenden Arten an:

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie Ihren KMS Schlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

Important


Sie können nur KMS Schlüssel verwenden, die in demselben aktiviert sind AWS-Region wie der Eimer. Wenn Sie „Aus Ihren KMS Schlüsseln auswählen“ wählen, listet die S3-Konsole nur 100 KMS Schlüssel pro Region auf. Wenn Sie mehr als 100 KMS Schlüssel in derselben Region haben, können Sie nur die ersten 100 KMS Schlüssel in der S3-Konsole sehen. Um einen KMS Schlüssel zu verwenden, der nicht in der Konsole aufgeführt ist, wählen Sie Enter AWS KMS key ARN, und geben Sie den KMS Schlüssel einARN.

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Weitere Informationen zur Verwendung von SSE - KMS mit Amazon S3 finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#). Weitere Informationen zur Verwendung von DSSE - KMS finden Sie unter [the section called "Zweischichtige serverseitige Verschlüsselung \(DSSE-\) KMS"](#).

- b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendet KMS, können Sie auch einen S3-Bucket-Key aktivieren. S3-Bucket Keys senken die Verschlüsselungskosten, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

 Note

S3-Bucket-Keys werden für DSSE - nicht unterstützt KMS.

8. Wählen Sie Änderungen speichern.

Verwendung der AWS CLI

Diese Beispiele zeigen Ihnen, wie Sie die Standardverschlüsselung mit SSE -S3 oder KMS mit SSE - mit einem S3-Bucket-Key konfigurieren.

Weitere Informationen zur Standardverschlüsselung finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#). Weitere Informationen zur Verwendung von AWS CLI Informationen zur Konfiguration der Standardverschlüsselung finden Sie unter [put-bucket-encryption](#).

Example — Standardverschlüsselung mit SSE -S3

In diesem Beispiel wird die Standardverschlüsselung von Buckets mit von Amazon S3 verwalteten Schlüsseln konfiguriert.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-configuration '{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "AES256"
      }
    }
  ]
}'
```

Example — Standardverschlüsselung mit SSE — KMS unter Verwendung eines S3-Bucket-Keys

In diesem Beispiel wird die Standard-Bucket-Verschlüsselung mit SSE - KMS unter Verwendung eines S3-Bucket-Keys konfiguriert.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-configuration '{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "KMS-Key-ARN"
      },
      "BucketKeyEnabled": true
    }
  ]
}'
```

Unter Verwendung des REST API

Verwenden Sie den REST API `PutBucketEncryption` Vorgang, um die Standardverschlüsselung zu aktivieren und den Typ der serverseitigen Verschlüsselung auf— SSE -S3KMS, SSE - oder DSSE - festzulegen. KMS

Weitere Informationen finden Sie unter [PutBucketEncryption](#) in der Amazon Simple Storage Service API Reference.

Überwachung der Standardverschlüsselung mit AWS CloudTrail und Amazon EventBridge

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon

S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Sie können Anfragen zur Standardverschlüsselungskonfiguration für Amazon S3 S3-Buckets verfolgen, indem Sie AWS CloudTrail Ereignisse. Die folgenden API Ereignisnamen werden in CloudTrail Protokollen verwendet:

- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

Sie können auch EventBridge Regeln erstellen, die den CloudTrail Ereignissen für diese API Aufrufe entsprechen. Weitere Informationen zu CloudTrail Ereignissen finden Sie unter [Aktivieren der Protokollierung für Objekte in einem Bucket mit der Konsole](#). Weitere Informationen zu EventBridge Ereignissen finden Sie unter [Ereignisse von AWS-Services](#).

Sie können CloudTrail Protokolle für Amazon S3-Aktionen auf Objektebene verwenden, um POST Anfragen an Amazon S3 nachzuverfolgenPUT. Sie können diese Aktionen verwenden, um zu überprüfen, ob die Standardverschlüsselung zum Verschlüsseln von Objekten verwendet wird, wenn eingehende Anfragen PUT keine Verschlüsselungs-Header haben.

Wenn Amazon S3 ein Objekt mit den Einstellungen der Standardverschlüsselung verschlüsselt, enthält das Protokoll eins der folgenden Felder als Name-Wert-Paar: "SSEApplied":"Default_SSE_S3", "SSEApplied":"Default_SSE_KMS" oder "SSEApplied":"Default_DSSE_KMS".

Wenn Amazon S3 ein Objekt mit den PUT-Verschlüsselungs-Headern verschlüsselt, enthält das Protokoll eins der folgenden Felder als Name-Wert-Paar: "SSEApplied":"SSE_S3", "SSEApplied":"SSE_KMS", "SSEApplied":"DSSE_KMS" oder "SSEApplied":"SSE_C".

Bei mehrteiligen Uploads sind diese Informationen in Ihren Betriebsanfragen enthalten. InitiateMultipartUpload API Weitere Hinweise zur Verwendung von CloudTrail und finden Sie CloudWatch unter. [Überwachen von Amazon S3](#)

Arbeiten mit Mountpoint für Amazon S3

Mountpoint für Amazon S3 ist ein Open-Source-Dateiclient mit hohem Durchsatz, um ein Amazon-S3-Bucket als lokales Dateisystem zu mounten. Mit Mountpoint können Ihre Anwendungen über Dateisystem-Operationen wie Öffnen und Lesen auf Objekte zugreifen, die in Amazon S3 gespeichert sind. Mountpoint übersetzt diese Operationen automatisch in API S3-Objekt-Aufrufe, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und Durchsatz von Amazon S3 zugreifen können.

Mountpoint für Amazon S3 ist [generell für den produktiven Einsatz in Ihren großen leseintensiven Anwendungen verfügbar](#): Data Lakes, Training für maschinelles Lernen, Rendern von Bildern, Simulation autonomer Fahrzeuge, Extrahieren, Transformieren und Laden (ETL) und mehr.

Mountpoint unterstützt grundlegende Dateisystem-Operationen und kann Dateien mit einer Größe von bis zu 5 TB lesen. Diese Lösung kann vorhandene Dateien auflisten und lesen sowie neue erstellen. Sie kann keine vorhandenen Dateien ändern oder Verzeichnisse löschen und unterstützt keine symbolischen Links oder Dateisperren. Mountpoint ist ideal für Anwendungen, die nicht alle Funktionen eines gemeinsam genutzten Dateisystems und POSIX Berechtigungen benötigen, sondern den elastischen Durchsatz von Amazon S3 zum Lesen und Schreiben großer S3-Datensätze benötigen. Einzelheiten finden Sie unter Verhalten des [Mountpoint-Dateisystems](#) auf GitHub. Für Workloads, die volle POSIX Unterstützung benötigen, empfehlen wir [Amazon FSx for Lustre und dessen Unterstützung für die Verknüpfung von S3-Buckets](#).

Mountpoint für Amazon S3 ist nur verfügbar für Linux Betriebssysteme. Sie können Mountpoint für den Zugriff auf S3-Objekte in allen Speicherklassen außer S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access Tier und S3 Intelligent-Tiering Deep Archive Access Tier verwenden.

Themen

- [Mountpoint installieren](#)
- [Mountpoint konfigurieren und verwenden](#)

Mountpoint installieren

Sie können vorgefertigte Pakete von Mountpoint für Amazon S3 über die Befehlszeile herunterladen und installieren. Die Anweisungen zum Herunterladen und Installieren von Mountpoint variieren je nachdem Linux Betriebssystem, das Sie verwenden.

Themen

- [RPMbasierte Distributionen \(Amazon Linux, Fedora, CentOS, RHEL\)](#)
- [DEBbasierte Distributionen \(Debian, Ubuntu\)](#)
- [Sonstige Linux Verteilungen](#)
- [Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3](#)

RPMbasierte Distributionen (Amazon Linux, Fedora, CentOS, RHEL)

1. Kopieren Sie den folgenden Download URL für Ihre Architektur.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm
```

2. Laden Sie das Mountpoint für Amazon-S3-Paket herunter. *download-link* Ersetzen Sie ihn durch den entsprechenden Download URL aus dem vorherigen Schritt.

```
wget download-link
```

3. (Optional) Überprüfen Sie die Authentizität und Integrität der heruntergeladenen Datei. Kopieren Sie zunächst die entsprechende Signatur URL für Ihre Architektur.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm.asc
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm.asc
```

Sehen Sie sich als Nächstes [Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3](#) an.

4. Installieren Sie das Paket mit dem folgenden Befehl:

```
sudo yum install ./mount-s3.rpm
```

5. Stellen Sie sicher, dass Mountpoint erfolgreich installiert wurde, indem Sie den folgenden Befehl eingeben:

```
mount-s3 --version
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
mount-s3 1.3.1
```

DEBbasierte Distributionen (Debian, Ubuntu)

1. Kopieren Sie den Download URL für Ihre Architektur.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb
```

2. Laden Sie das Mountpoint für Amazon-S3-Paket herunter. *download-link* Ersetzen Sie ihn durch den entsprechenden Download URL aus dem vorherigen Schritt.

```
wget download-link
```

3. (Optional) Überprüfen Sie die Authentizität und Integrität der heruntergeladenen Datei. Kopieren Sie zunächst die Signatur URL für Ihre Architektur.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb.asc
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb.asc
```

Sehen Sie sich als Nächstes [Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3](#) an.

4. Installieren Sie das Paket mit dem folgenden Befehl:

```
sudo apt-get install ./mount-s3.deb
```

5. Stellen Sie sicher, dass Mountpoint für Amazon S3 erfolgreich installiert wurde, indem Sie den folgenden Befehl ausführen:

```
mount-s3 --version
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
mount-s3 1.3.1
```

Sonstige Linux Verteilungen

1. Schlagen Sie in der Dokumentation Ihres Betriebssystems nach, um die erforderlichen Pakete FUSE und libfuse2 zu installieren.
2. Kopieren Sie den Download URL für Ihre Architektur.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz
```

3. Laden Sie das Mountpoint für Amazon-S3-Paket herunter. *download-link* Ersetzen Sie ihn durch den entsprechenden Download URL aus dem vorherigen Schritt.

```
wget download-link
```

- (Optional) Überprüfen Sie die Authentizität und Integrität der heruntergeladenen Datei. Kopieren Sie zunächst die Signatur URL für Ihre Architektur.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz.asc
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz.asc
```

Sehen Sie sich als Nächstes [Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3](#) an.

- Installieren Sie das Paket mit dem folgenden Befehl:

```
sudo mkdir -p /opt/aws/mountpoint-s3 && sudo tar -C /opt/aws/mountpoint-s3 -xzf ./mount-s3.tar.gz
```

- Fügen Sie das mount-s3-Binary Ihrer PATH-Umgebungsvariablen hinzu. Fügen Sie in Ihrer \$HOME/.profile-Datei die folgende Zeile an:

```
export PATH=$PATH:/opt/aws/mountpoint-s3/bin
```

Speichern Sie die .profile-Datei und führen Sie den folgenden Befehl aus:

```
source $HOME/.profile
```

- Stellen Sie sicher, dass Mountpoint für Amazon S3 erfolgreich installiert wurde, indem Sie den folgenden Befehl ausführen:

```
mount-s3 --version
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
mount-s3 1.3.1
```

Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3

1. Installieren GnuPG (der `gpg` Befehl). Es ist erforderlich, um die Authentizität und Integrität eines heruntergeladenen Mountpoint for Amazon S3-Pakets zu überprüfen. GnuPG ist standardmäßig auf Amazon Linux Amazon Machine Images (AMIs) installiert. Nach der Installation GnuPG, fahren Sie mit Schritt 2 fort.
2. Laden Sie den öffentlichen Mountpoint-Schlüssel herunter, indem Sie den folgenden Befehl ausführen:

```
wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
```

3. Importieren Sie den öffentlichen Mountpoint-Schlüssel in Ihren Schlüsselbund, indem Sie den folgenden Befehl ausführen:

```
gpg --import KEYS
```

4. Überprüfen Sie den Fingerabdruck des öffentlichen Mountpoint-Schlüssels, indem Sie den folgenden Befehl ausführen:

```
gpg --fingerprint mountpoint-s3@amazon.com
```

Vergewissern Sie sich, dass die angezeigte Fingerabdruck-Zeichenfolge mit der folgenden Zeichenfolge übereinstimmt:

```
673F E406 1506 BB46 9A0E F857 BE39 7A52 B086 DA5A
```

Wenn die Zeichenfolge für den Fingerabdruck nicht übereinstimmt, beenden Sie die Installation von Mountpoint nicht und wenden Sie sich an [AWS Support](#).

5. Laden Sie die Paket-Signaturdatei herunter. Ersetzen Sie *signature-link* durch den entsprechenden Signaturlink aus den vorherigen Abschnitten.

```
wget signature-link
```

6. Überprüfen Sie die Signatur des heruntergeladenen Pakets, indem Sie den folgenden Befehl ausführen. Ersetzen Sie *signature-filename* durch den Dateinamen aus dem vorherigen Schritt.

```
gpg --verify signature-filename
```

Geben Sie beispielsweise bei RPM basierten Distributionen, einschließlich Amazon Linux, den folgenden Befehl ein:

```
gpg --verify mount-s3.rpm.asc
```

7. Die Ausgabe sollte den Text `Good signature` enthalten. Wenn die Ausgabe den Text `BAD signature` enthält, laden Sie die Mountpoint-Paketdatei erneut herunter und wiederholen Sie diese Schritte. Wenn das Problem weiterhin besteht, beenden Sie die Installation von Mountpoint nicht und wenden Sie sich an [AWS Support](#).

Die Ausgabe kann eine Warnung über eine vertrauenswürdige Signatur enthalten. Dies deutet nicht auf ein Problem hin. Es bedeutet nur, dass Sie den öffentlichen Mountpoint-Schlüssel nicht unabhängig verifiziert haben.

Mountpoint konfigurieren und verwenden

Um Mountpoint für Amazon S3 verwenden zu können, benötigt Ihr Host eine gültige AWS Anmeldeinformationen mit Zugriff auf den Bucket oder die Buckets, die Sie mounten möchten. Informationen zu verschiedenen Authentifizierungsmethoden finden Sie unter Mountpoint [AWS Anmeldeinformationen ein](#) GitHub.

Sie können beispielsweise ein neues erstellen AWS Identity and Access Management (IAM) Benutzer und Rolle für diesen Zweck. Stellen Sie sicher, dass diese Rolle Zugriff auf den Bucket oder die Buckets hat, die Sie mounten möchten. Sie können [die IAM Rolle mit einem EC2 Instance-Profil an Ihre Amazon-Instance übergeben](#).

Verwenden von Mountpoint für Amazon S3

Verwenden Sie Mountpoint für Amazon S3, um Folgendes zu tun:

1. Mounten Sie Buckets mit dem `mount -s3`-Befehl.

Ersetzen Sie im folgenden Beispiel `DOC-EXAMPLE-BUCKET` durch den Namen Ihres S3-Buckets und ersetzen Sie `~/mnt` durch das Verzeichnis auf Ihrem Host, in dem Ihr S3-Bucket gemountet werden soll.

```
mkdir ~/mnt  
mount-s3 DOC-EXAMPLE-BUCKET ~/mnt
```

Da der Mountpoint-Client standardmäßig im Hintergrund ausgeführt wird, bietet Ihnen das `~/mnt`-Verzeichnis jetzt Zugriff auf die Objekte in Ihrem S3-Bucket.

- Greifen Sie über Mountpoint auf die Objekte in Ihrem Bucket zu.

Nachdem Sie Ihren Bucket lokal bereitgestellt haben, können Sie Common verwenden Linux Befehle wie `cat` oder `ls`, um mit Ihren S3-Objekten zu arbeiten. Mountpoint für Amazon S3 interpretiert Schlüssel in Ihrem S3-Bucket als Dateisystempfade. Dazu werden sie anhand des Schrägstrichs (`/`) getrennt. Wenn Sie zum Beispiel den Objektschlüssel `Data/2023-01-01.csv` in Ihrem Bucket haben, finden Sie ein Verzeichnis mit dem Namen `Data` in Ihrem Mountpoint-Dateisystem mit einer Datei namens `2023-01-01.csv` darin.

Mountpoint for Amazon S3 implementiert bewusst nicht die vollständige [POSIX](#) Standardspezifikation für Dateisysteme. Mountpoint ist für Workloads optimiert, die über eine Dateisystemschnittstelle Lese- und Schreibzugriff mit hohem Durchsatz auf Daten benötigen, die in Amazon S3 gespeichert aber ansonsten nicht auf Dateisystemfunktionen angewiesen sind. Weitere Informationen finden Sie unter [Dateisystemverhalten](#) von Mountpoint for Amazon S3 auf GitHub. Kunden, die eine umfassendere Dateisystemsemantik benötigen, sollten andere in Betracht ziehen AWS Dateidienste wie [Amazon Elastic File System \(AmazonEFS\)](#) oder [Amazon FSx](#).

- Heben Sie das Mounting Ihres Bucket mit dem `umount`-Befehl auf. Mit diesem Befehl wird Mounting Ihres S3-Bucket aufgehoben und Mountpoint wird beendet.

Um den folgenden Beispielbefehl zu verwenden, ersetzen Sie `~/mnt` durch das Verzeichnis auf Ihrem Host, in dem Ihr S3-Bucket gemountet ist.

```
umount ~/mnt
```

Note

Wenn Sie eine Liste der Optionen für diesen Befehl erhalten möchten, führen Sie `umount --help` aus.

Weitere Informationen zur Mountpoint-Konfiguration finden Sie unter [S3-Bucket-Konfiguration](#) und [Dateisystemkonfiguration](#) auf GitHub.

Konfigurieren von Caching in Mountpoint

Wenn Sie Mountpoint für Amazon S3 verwenden, können Sie es so konfigurieren, dass die zuletzt abgerufenen Daten aus Ihren S3-Buckets im EC2 Amazon-Instance-Speicher oder einem angehängten Amazon-Volume zwischengespeichert werden. Wenn Sie diese Daten im Cache speichern, kann dies die Leistung beschleunigen und die Kosten für wiederholte Datenzugriffe senken. Das Caching in Mountpoint ist ideal für Anwendungsfälle, in denen Sie wiederholt dieselben Daten lesen, die sich während der mehrfachen Lesevorgänge nicht ändern. Sie können das Caching beispielsweise bei Machine-Learning-Trainingsaufgaben verwenden, bei denen ein Trainingsdatensatz mehrmals gelesen werden muss, um die Modellgenauigkeit zu verbessern.

Wenn Sie einen S3-Bucket bereitstellen, können Sie optional das Caching mittels Flags aktivieren. Sie können den Speicherort und die Größe des Datencache sowie die Dauer der Beibehaltung von Metadaten im Cache konfigurieren. Wenn Sie einen Bucket bereitstellen und Caching aktiviert ist, erstellt Mountpoint ein leeres Unterverzeichnis am konfigurierten Cache-Speicherort, wenn dieses Unterverzeichnis noch nicht vorhanden ist. Wenn Sie einen Bucket bereitstellen und diese Bereitstellung dann aufheben, löscht Mountpoint den Inhalt des Cache-Speicherorts. Weitere Informationen zur Konfiguration und Verwendung von Caching in Mountpoint finden Sie unter [Mountpoint for Amazon S3 Caching configuration on GitHub](#).

Wenn Sie einen S3-Bucket bereitstellen, können Sie optional das Caching mittels des Flags `--cache` `CACHE_PATH` aktivieren. Ersetzen Sie im folgenden Beispiel `CACHE_PATH` durch den Dateipfad zu dem Verzeichnis, in dem Sie Ihre Daten zwischenspeichern möchten. Ersetzen `DOC-EXAMPLE-BUCKET` mit dem Namen Ihres S3-Buckets und ersetzen `~/mnt` mit dem Verzeichnis auf Ihrem Host, in dem Ihr S3-Bucket gemountet werden soll.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH DOC-EXAMPLE-BUCKET ~/mnt
```

Important

Wenn Sie Caching aktivieren, speichert Mountpoint unverschlüsselte Objekthinhalte aus Ihrem S3-Bucket an dem bei der Bereitstellung konfigurierten Caching-Speicherort. Zum Schutz Ihrer Daten sollten den Zugriff auf den Datencache-Speicherort einschränken.

Fehlerbehebung bei Mountpoint

Mountpoint für Amazon S3 wird unterstützt von AWS Support. Wenn Sie Hilfe benötigen, wenden Sie sich an [AWS Support Zentrum](#).

[Sie können Mountpoint-Probleme auch unter überprüfen und einreichen](#) GitHub.

Wenn Sie in diesem Projekt ein potenzielles Sicherheitsproblem entdecken, bitten wir Sie, uns darüber zu informieren AWS Sicherheit finden Sie auf unserer [Seite zur Meldung von Sicherheitslücken](#). Erstellen Sie keine öffentliche GitHub Problem.

Wenn sich Ihre Anwendung mit Mountpoint unerwartet verhält, können Sie Ihre Protokollinformationen überprüfen, um das Problem zu diagnostizieren.

Protokollierung

Standardmäßig sendet Mountpoint Protokollinformationen mit hohem Schweregrad an [syslog](#).

Um Protokolle auf den modernsten Geräten anzuzeigen Linux Distributionen, einschließlich Amazon Linux, führen den folgenden `journal` Befehl aus:

```
journalctl -e SYSLOG_IDENTIFIER=mount-s3
```

Auf anderen Linux Auf Systemen werden `syslog` Einträge wahrscheinlich in eine Datei wie `geschrieben/var/log/syslog`.

Sie können diese Protokolle verwenden, um Fehler in Ihrer Anwendung zu beheben. Wenn Ihre Anwendung beispielsweise versucht, eine vorhandene Datei zu überschreiben, schlägt der Vorgang fehl und im Protokoll wird eine Zeile angezeigt, die der folgenden ähnelt:

```
[WARN] open{req=12 ino=2}: mountpoint_s3::fuse: open failed: inode error: inode 2 (full key "README.md") is not writable
```

[Weitere Informationen finden Sie unter Mountpoint for Amazon S3 Log-in](#) GitHub.

Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration ist eine Funktion auf Bucket-Ebene, die die schnelle, einfache und sichere Übertragung von Dateien über große Entfernungen zwischen Ihrem Client und einem

S3-Bucket ermöglicht. Transfer Acceleration wurde entwickelt, um Übertragungsgeschwindigkeiten aus der ganzen Welt in S3-Buckets zu optimieren. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte von Amazon CloudFront. Sobald die Daten an einem Edge-Standort eingeht, werden sie über einen optimierten Netzwerkpfad an Ihren Amazon S3-Bucket weitergeleitet.

Wenn Sie Transfer Acceleration verwenden, fallen möglicherweise zusätzliche Gebühren für die Datenübertragung an. Weitere Informationen zu Preisen finden Sie unter [Amazon S3-Preise](#).

Gründe für die Nutzung von Amazon S3 Transfer Acceleration

Es gibt verschiedene Gründe für die Verwendung von Transfer Acceleration auf einem Bucket:

- Ihre Kunden laden in einen zentralen Bucket aus der ganzen Welt hoch.
- Sie übertragen regelmäßig mehrere Gigabyte bis Terabyte von Daten über mehrere Kontinente hinweg.
- Sie können beim Hochladen in Amazon S3 nicht die gesamte verfügbare Bandbreite über das Internet nutzen.

Weitere Informationen darüber, wann Sie Transfer Acceleration verwenden sollten, finden Sie unter [Amazon S3 FAQs](#).

Anforderungen für die Verwendung von Transfer Acceleration

Folgendes ist erforderlich, wenn Sie Transfer Acceleration für einen S3-Bucket verwenden:

- Transfer Acceleration wird nur bei virtuell gehosteten Anfragen unterstützt. Weitere Informationen zu virtuell gehosteten Anfragen finden Sie unter [Anfragen stellen mit dem REST API](#).
- Der Name des für Transfer Acceleration verwendeten Buckets muss DNS -konform sein und darf keine Punkte („.“) enthalten.
- Transfer Acceleration muss im Bucket aktiviert sein. Weitere Informationen finden Sie unter [Aktivieren und Verwenden von S3 Transfer Acceleration](#).

Nach der Aktivierung von Transfer Acceleration für einen Bucket kann es bis zu 20 Minuten dauern, bis sich die Datenübertragungsgeschwindigkeit in den Bucket erhöht.

Note

Transfer Acceleration wird zurzeit für Buckets in den folgenden Regionen unterstützt:

- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Kanada (Zentral): (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Südamerika (São Paulo) (sa-east-1)
- USA Ost (Nord-Virginia): (us-east-1)
- USA Ost (Ohio): (us-east-2)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)

- Um auf den Bucket zuzugreifen, der für Transfer Acceleration konfiguriert ist, müssen Sie den Endpunkt `bucketname.s3-accelerate.amazonaws.com` verwenden. Oder verwenden Sie den Dual-Stack-Endpunkt `bucketname.s3-accelerate.dualstack.amazonaws.com`, um eine Verbindung zum aktivierten Bucket herzustellen. IPv6 Sie können weiterhin die regulären Endpunkte für die Standarddatenübertragung verwenden.
- Sie müssen der Bucket-Eigentümer sein, um den Transfer Acceleration-Status festlegen zu können. Der Bucket-Eigentümer kann anderen Benutzern Berechtigungen erteilen, um ihnen zu gestatten, den Beschleunigungsstatus für einen Bucket einzurichten. Die `s3:PutAccelerateConfiguration`-Berechtigung gestattet Benutzern, Transfer Acceleration für einen Bucket zu aktivieren oder zu deaktivieren. Die `s3:GetAccelerateConfiguration` Berechtigung ermöglicht es Benutzern, den Status Transfer Acceleration eines Buckets zurückzugeben, der entweder `Enabled` oder `Suspended` ist.

In den folgenden Abschnitten wird beschrieben, wie Sie beginnen und Amazon S3 Transfer Acceleration für die Übertragung von Daten verwenden.

Themen

- [Erste Schritte mit Amazon S3 Transfer Acceleration](#)
- [Aktivieren und Verwenden von S3 Transfer Acceleration](#)
- [Verwenden des Amazon S3 Transfer Acceleration Speed Comparison-Tools](#)

Erste Schritte mit Amazon S3 Transfer Acceleration

Sie können Amazon S3 Transfer Acceleration für die schnelle, einfache und sichere Übertragung von Dateien über große Entfernungen zwischen Ihrem Client und einem S3-Bucket nutzen. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte in Amazon CloudFront. Sobald die Daten an einem Edge-Standort eingeht, werden sie über einen optimierten Netzwerkpfad an Ihren Amazon S3-Bucket weitergeleitet.

Um mit der Verwendung von Amazon S3 Transfer Acceleration zu beginnen, führen Sie die folgenden Schritte aus:

1. Aktivieren der Transfer Acceleration auf einem Bucket

Sie können Transfer Acceleration für einen Bucket auf eine der folgenden Arten aktivieren:

- Verwenden der Amazon S3-Konsole
- Verwenden Sie den REST API [PUTBucket Accelerate-Vorgang](#).
- Verwenden Sie den AWS CLI and AWS SDKs. Weitere Informationen finden Sie unter [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Weitere Informationen finden Sie unter [Aktivieren und Verwenden von S3 Transfer Acceleration](#).

Note

Damit Ihr Bucket mit der Übertragungsbeschleunigung funktioniert, muss der Bucket-Name den DNS Benennungsanforderungen entsprechen und darf keine Punkte („.“) enthalten.

2. Übertragen von Daten in den und aus dem beschleunigungsfähigen Bucket

Verwenden Sie einen der folgenden s3-beschleunigten Endpunkt-Domännennamen:

- Um auf einen beschleunigungsfähigen Bucket zuzugreifen, nutzen Sie *bucketname*.s3-accelerate.amazonaws.com.

- Um auf einen Bucket mit aktivierter Beschleunigung zuzugreifen, verwenden Sie IPv6 `bucketname.s3-accelerate.dualstack.amazonaws.com`

Amazon S3 S3-Dual-Stack-Endpunkte unterstützen Anfragen an S3-Buckets über und. IPv6 IPv4 Der Transfer Acceleration-Dual-Stack-Endpunkt verwendet nur den virtuellen Hosting-Endpunktnamen. Weitere Informationen erhalten Sie unter [Erste Schritte für Anforderungen über IPv6](#) und [Verwenden von Amazon-S3-Dual-Stack-Endpunkten](#).

Note

Ihre Anwendung zur Datenübertragung muss eine der beiden folgenden Endpunkt-Typen verwenden, um auf den Bucket zuzugreifen und die Datenübertragung beschleunigen zu können: `.s3-accelerate.amazonaws.com` oder `.s3-accelerate.dualstack.amazonaws.com` für den Dual-Stack-Endpunkt. Wenn Sie die Standarddatenübertragung verwenden möchten, können Sie weiterhin die regulären Endpunkte nutzen.

Sie können Ihr Amazon S3 PUT GET S3-Objekt und Ihre Objektanforderungen auf den Domainnamen des `s3-accelerate` Endpunkts verweisen, nachdem Sie Transfer Acceleration aktiviert haben. Nehmen wir zum Beispiel an, Sie haben derzeit eine REST API Anwendung, die [PUTObject](#) verwendet und den Hostnamen `mybucket.s3.us-east-1.amazonaws.com` in der PUT Anfrage verwendet. Um PUT zu beschleunigen, ändern Sie den Hostnamen in Ihrer Anfrage in `mybucket.s3-accelerate.amazonaws.com`. Um wieder die Standard-Upload-Geschwindigkeit zu verwenden, ändern Sie den Namen zurück in `mybucket.s3.us-east-1.amazonaws.com`.

Nachdem Transfer Acceleration aktiviert wurde, kann es bis zu 20 Minuten dauern, bis Sie den Leistungsvorteil wahrnehmen. Der beschleunigte Endpunkt steht zur Verfügung, sobald Sie Transfer Acceleration aktivieren.

Sie können den Accelerate-Endpunkt in der AWS CLI, AWS SDKs und andere Tools, die Daten zu und von Amazon S3 übertragen. Wenn Sie das verwenden AWS SDKs, verwenden einige der unterstützten Sprachen ein Accelerate-Endpoint-Client-Konfigurationsflag, sodass Sie den Endpunkt für Transfer Acceleration nicht explizit auf festlegen müssen `bucketname.s3-accelerate.amazonaws.com`. Beispiele für das Flag für eine Client-Konfiguration mit beschleunigtem Endpunkt finden Sie unter [Aktivieren und Verwenden von S3 Transfer Acceleration](#).

Sie können alle Amazon S3-Vorgänge über die Endpunkte der Transfer Acceleration mit Ausnahme der Folgenden verwenden:

- [GETService \(Buckets auflisten\)](#)
- [PUTBucket \(Bucket erstellen\)](#)
- [DELETEEimer](#)

Außerdem unterstützt Amazon S3 Transfer Acceleration keine regionsübergreifenden Kopien mit [PUTObject — Copy](#).

Aktivieren und Verwenden von S3 Transfer Acceleration

Sie können Amazon S3 Transfer Acceleration verwenden, um Dateien schnell und sicher über große Entfernungen zwischen Ihrem Client und einem S3-Bucket zu übertragen. Sie können Transfer Acceleration über die S3-Konsole aktivieren, AWS Command Line Interface (AWS CLI)API, oder die AWS SDKs.

Dieser Abschnitt bietet Beispiele für die Aktivierung von Amazon S3 Transfer Acceleration und die Verwendung des beschleunigten Endpunkts für den aktivierten Bucket.

Weitere Informationen zu den Anforderungen für Transfer Acceleration finden Sie unter [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#).

Verwenden der S3-Konsole

Note

Wenn Sie beschleunigte und nicht beschleunigte Upload-Geschwindigkeiten vergleichen möchten, öffnen Sie das [Amazon S3 Transfer Acceleration Speed Comparison-Tool](#). Das Tool zum Geschwindigkeitsvergleich verwendet einen mehrteiligen Upload, um eine Datei von Ihrem Browser in verschiedene zu übertragen AWS-Regionen mit und ohne Amazon S3 S3-Übertragungsbeschleunigung. Sie können die Upload-Geschwindigkeit für direkte Uploads vergleichen und beschleunigte Uploads nach Region übertragen.

Die Transfer Acceleration für einen S3-Bucket aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Transfer Acceleration aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Transfer acceleration (Beschleunigung übertragen) die Option Edit (Bearbeiten) aus.
5. Wählen Sie Enable (Aktivieren) und wählen Sie Save changes (Änderungen speichern) aus.

Zugriff auf beschleunigte Datenübertragungen

1. Nachdem Amazon S3 die Übertragungsbeschleunigung für Ihren Bucket aktiviert hat, sehen Sie sich den Tab Properties (Eigenschaften) für den Bucket an.
2. Unter Transfer acceleration (Übertragungsbeschleunigung) wird unter Accelerated endpoint (beschleunigter Endpunkt) der Endpunkt der Übertragungsbeschleunigung für Ihren Bucket angezeigt. Verwenden Sie diesen Endpunkt, um auf beschleunigte Datenübertragungen von und zu Ihrem Bucket zuzugreifen.

Wenn Sie die Transfer Acceleration aussetzen, funktioniert der beschleunigte Endpunkt nicht mehr.

Verwendung der AWS CLI

Im Folgenden finden Sie Beispiele für AWS CLI Befehle, die für die Übertragungsbeschleunigung verwendet werden. Für Anweisungen zur Einrichtung des AWS CLI, finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

Aktivieren der Transfer Acceleration auf einem Bucket

Verwenden Sie die AWS CLI [put-bucket-accelerate-configuration](#) Befehl, um die Übertragungsbeschleunigung für einen Bucket zu aktivieren oder zu unterbrechen.

Im folgenden Beispiel wird `Status=Enabled` zur Aktivierung von Transfer Acceleration für einen Bucket festgelegt. Sie verwenden `Status=Suspended`, um Transfer Acceleration auszusetzen.

Example

```
$ aws s3api put-bucket-accelerate-configuration --bucket bucketname --accelerate-configuration Status=Enabled
```


Nutzen der Transfer Acceleration

Sie können alle Amazon S3 S3-Anfragen von `s3` und `s3api` weiterleiten AWS CLI Befehle an den Accelerate-Endpunkt: `s3-accelerate.amazonaws.com`. Stellen Sie dazu den Konfigurationswert `use_accelerate_endpoint true` in einem Profil in Ihrer AWS Konfigurationsdatei. Transfer Acceleration muss für Ihren Bucket aktiviert sein, um den beschleunigten Endpunkt nutzen zu können.

Alle Anforderungen werden mit der virtuellen Bucket-Adressierung gesendet: `my-bucket.s3-accelerate.amazonaws.com`. `ListBuckets`-, `CreateBucket`-, und `DeleteBucket`-Anfragen werden nicht an den beschleunigten Endpunkt gesendet, da der Endpunkt diese Vorgänge nicht unterstützt.

Weitere Informationen über `use_accelerate_endpoint` finden Sie unter [AWS CLI S3-Konfiguration](#) im AWS CLI Befehlsreferenz.

Das folgende Beispiel setzt `use_accelerate_endpoint` im Standardprofil auf `true`.

Example

```
$ aws configure set default.s3.use_accelerate_endpoint true
```

Wenn Sie den Accelerate-Endpunkt für einige verwenden möchten AWS CLI Befehle, andere jedoch nicht, können Sie eine der beiden folgenden Methoden verwenden:

- Verwenden Sie den beschleunigten Endpunkt für jeden `s3`- oder `s3api`-Befehl, indem Sie den Parameter `--endpoint-url` auf `https://s3-accelerate.amazonaws.com` setzen.
- Richten Sie separate Profile in Ihrer ein AWS Konfigurationsdatei. Legen Sie beispielsweise ein Profil an, das `use_accelerate_endpoint` auf `true` setzt, und ein Profil, das `use_accelerate_endpoint` nicht setzt. Wenn Sie einen Befehl ausführen, geben Sie an, welches Profil Sie verwenden möchten, abhängig davon, ob Sie den beschleunigten Endpunkt verwenden möchten.

Hochladen eines Objekts in einen Bucket, der für Transfer Acceleration geeignet ist

Das folgende Beispiel lädt eine Datei in einen Bucket hoch, der für Transfer Acceleration konfiguriert ist. Dazu verwendet es das Standardprofil, das für die Verwendung des beschleunigten Endpunkts konfiguriert wurde.

Example

```
$ aws s3 cp file.txt s3://bucketname/keyname --region region
```

Das folgende Beispiel lädt eine Datei in einen Bucket hoch, der für Transfer Acceleration konfiguriert ist. Dazu verwendet es den Parameter `--endpoint-url` für die Angabe des beschleunigten Endpunkts.

Example

```
$ aws configure set s3.addressing_style virtual
$ aws s3 cp file.txt s3://bucketname/keyname --region region --endpoint-url https://s3-  
accelerate.amazonaws.com
```

Verwendung der AWS SDKs

Im Folgenden finden Sie Beispiele für die Verwendung von Transfer Acceleration zum Hochladen von Objekten auf Amazon S3 mithilfe von AWS SDK. Einige der AWS SDK-unterstützte Sprachen (zum Beispiel Java und .NET) verwenden Sie ein Client-Konfigurationsflag für Accelerate Endpoint, sodass Sie den Endpunkt für Transfer Acceleration nicht explizit aufsetzen müssen `bucketname.s3-accelerate.amazonaws.com`.

Java

Example

Das folgende Beispiel zeigt, wie Sie einen beschleunigten Endpunkt für das Hochladen eines Objekts in Amazon S3 verwenden. Das Beispiel erledigt Folgendes:

- Erstellt einen `AmazonS3Client`, der für die Verwendung von beschleunigten Endpunkten konfiguriert ist. Für alle Buckets, auf die der Client zugreift, muss Transfer Acceleration aktiviert sein.
- Aktiviert Transfer Acceleration auf einem angegebenen Bucket. Dieser Schritt ist nur erforderlich, wenn für den von Ihnen angegebenen Bucket nicht bereits Transfer Acceleration aktiviert ist.
- Überprüft, ob Transfer Acceleration für den angegebenen Bucket aktiviert ist.
- Lädt ein neues Objekt in den angegebenen Bucket hoch und verwendet dazu den beschleunigten Endpunkt des Buckets.

Weitere Informationen zur Verwendung von Transfer Acceleration finden Sie unter [Erste Schritte mit Amazon S3 Transfer Acceleration](#). [Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketAccelerateConfiguration;
import com.amazonaws.services.s3.model.BucketAccelerateStatus;
import com.amazonaws.services.s3.model.GetBucketAccelerateConfigurationRequest;
import com.amazonaws.services.s3.model.SetBucketAccelerateConfigurationRequest;

public class TransferAcceleration {
    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";

        try {
            // Create an Amazon S3 client that is configured to use the accelerate
            endpoint.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .enableAccelerateMode()
                .build();

            // Enable Transfer Acceleration for the specified bucket.
            s3Client.setBucketAccelerateConfiguration(
                new SetBucketAccelerateConfigurationRequest(bucketName,
                    new BucketAccelerateConfiguration(
                        BucketAccelerateStatus.Enabled)));

            // Verify that transfer acceleration is enabled for the bucket.
            String accelerateStatus = s3Client.getBucketAccelerateConfiguration(
                new GetBucketAccelerateConfigurationRequest(bucketName))
                .getStatus();

            System.out.println("Bucket accelerate status: " + accelerateStatus);
        }
    }
}
```

```
        // Upload a new object using the accelerate endpoint.
        s3Client.putObject(bucketName, keyName, "Test object for transfer
acceleration");
        System.out.println("Object \"" + keyName + "\" uploaded with transfer
acceleration.");
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Das folgende Beispiel zeigt, wie Sie AWS SDK für .NET um Transfer Acceleration für einen Bucket zu aktivieren. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TransferAccelerationTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
```

```
s3Client = new AmazonS3Client(bucketRegion);
EnableAccelerationAsync().Wait();
}

static async Task EnableAccelerationAsync()
{
    try
    {
        var putRequest = new PutBucketAccelerateConfigurationRequest
        {
            BucketName = bucketName,
            AccelerateConfiguration = new AccelerateConfiguration
            {
                Status = BucketAccelerateStatus.Enabled
            }
        };
        await
s3Client.PutBucketAccelerateConfigurationAsync(putRequest);

        var getRequest = new GetBucketAccelerateConfigurationRequest
        {
            BucketName = bucketName
        };
        var response = await
s3Client.GetBucketAccelerateConfigurationAsync(getRequest);

        Console.WriteLine("Acceleration state = '{0}' ",
response.Status);
    }
    catch (AmazonS3Exception amazonS3Exception)
    {
        Console.WriteLine(
"Error occurred. Message:'{0}' when setting transfer
acceleration",
amazonS3Exception.Message);
    }
}
}
```

Wenn Sie ein Objekt in einen Bucket hochladen, für den Transfer Acceleration aktiviert ist, verwenden Sie für die Angabe des beschleunigten Endpunkts den Zeitpunkt, an dem ein Client erstellt wurde.

```
var client = new AmazonS3Client(new AmazonS3Config
    {
        RegionEndpoint = TestRegionEndpoint,
        UseAccelerateEndpoint = true
    })
```

Javascript

Ein Beispiel für die Aktivierung von Transfer Acceleration mit dem AWS SDK finden Sie JavaScript unter [Aufrufen des putBucketAccelerate Konfigurationsvorgangs](#) in der AWS SDK als JavaScript API Referenz.

Python (Boto)

Ein Beispiel für die Aktivierung von Transfer Acceleration mithilfe von SDK for Python finden Sie unter [put_bucket_accelerate_configuration](#) in der AWS SDK für Python (Boto3) API -Referenz.

Other

Für Informationen zur Verwendung anderer AWS SDKs, siehe [Beispielcode und Bibliotheken](#).

Mit dem REST API

Verwenden Sie den REST API PutBucketAccelerateConfiguration Vorgang, um die beschleunigte Konfiguration für einen vorhandenen Bucket zu aktivieren.

Weitere Informationen finden Sie [PutBucketAccelerateConfiguration](#) in der Amazon Simple Storage Service API Reference.

Verwenden des Amazon S3 Transfer Acceleration Speed Comparison-Tools

Sie können das [Amazon S3 Transfer Acceleration Speed Comparison Tool](#) verwenden, um beschleunigte und nicht beschleunigte Upload-Geschwindigkeiten in allen Amazon S3-Regionen zu vergleichen. Das Speed Comparison-Tool verwendet mehrteilige Uploads, um eine Datei von Ihrem Browser in verschiedene Amazon S3-Regionen mit und ohne die Verwendung von Transfer Acceleration zu übertragen.

Sie können mit einer der folgenden Methoden auf das Speed Comparison-Tool zugreifen:

- Kopieren Sie den folgenden Text URL in Ihr Browserfenster und ersetzen Sie *region* mit dem AWS-Region die Sie verwenden (zum Beispielus-west-2) und *yourBucketName* mit dem Namen des Buckets, den Sie auswerten möchten:

```
https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparison.html?region=region&origBucketName=yourBucketName
```

Eine Liste der von Amazon S3 unterstützten Regionen finden Sie unter [Amazon S3-Endpunkte und Kontingente](#) in der Allgemeine AWS-Referenz.

- Verwenden der Amazon S3-Konsole

Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung

Im Allgemeinen zahlen Bucket-Eigentümer alle Amazon-S3-Speicher- und -Datenübertragungskosten, die ihrem Bucket zuzuordnen sind. Sie können jedoch einen Bucket als Bucket mit Zahlung durch den Anforderer konfigurieren. Bei Buckets mit Zahlung durch den Anforderer zahlt der Auftraggeber statt des Bucket-Eigentümers die Kosten für die Anforderung und den Daten-Download aus dem Bucket. Der Bucket-Eigentümer zahlt immer die Kosten für das Speichern der Daten.

In der Regel konfigurieren Sie Buckets dann als Buckets mit Zahlung durch den Anforderer, wenn Sie Daten freigeben möchten, aber keine Gebühren dafür übernehmen wollen, wenn andere auf die Daten zugreifen. Sie können beispielsweise Buckets mit Zahlung durch den Anforderer verwenden, wenn Sie große Datasets bereitstellen, beispielsweise Postleitzahlenverzeichnisse, Referenzdaten, Geodateninformationen oder Web-Crawling-Daten.

Important

Wenn Sie die Zahlung durch den Anforderer für einen Bucket aktivieren, ist kein anonymer Zugriff auf den Bucket zulässig.

Sie müssen alle Anforderungen für Buckets mit Zahlung durch den Anforderer authentifizieren. Die Anfrageauthentifizierung ermöglicht Amazon S3, den Auftraggeber zu identifizieren und ihm seine Verwendung des Buckets mit Zahlung durch den Anforderer in Rechnung zu stellen.

Wenn der Anforderer davon ausgeht AWS Identity and Access Management (IAM) Rolle, bevor sie ihre Anfrage stellt, wird die Anfrage dem Konto, zu dem die Rolle gehört, in Rechnung gestellt. Weitere Informationen zu IAM [IAMRollen](#) finden Sie im IAMBenutzerhandbuch unter Rollen.

Nachdem Sie einen Bucket als Bucket für „Requester Pays“ konfiguriert haben, müssen die Anforderer nachweisen, dass sie sich darüber im Klaren sind, dass ihnen die Anfrage und der Datendownload in Rechnung gestellt werden. Um nachzuweisen, dass sie die Gebühren akzeptieren, müssen die Anforderer entweder `x-amz-request-payer` als Header in ihre API Anfrage für DELETE, GET HEADPOST, und PUT -Anfragen aufnehmen oder den `RequestPayer` Parameter in ihrer Anfrage hinzufügen. REST Für CLI Anfragen können Anforderer den `--request-payer` Parameter verwenden.

Example — Die Verwendung von Requester Pays beim Löschen eines Objekts

Um Folgendes zu verwenden [DeleteObjectVersion](#) API Ersetzen Sie beispielsweise die *user input placeholders* durch Ihre eigenen Informationen.

```
DELETE /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-mfa: MFA
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention
x-amz-expected-bucket-owner: ExpectedBucketOwner
```

Wenn der Anforderer Objekte mithilfe der wiederherstellt [RestoreObject](#) API, Requester Pays wird unterstützt, solange der `x-amz-request-payer` Header oder der `RequestPayer` Parameter in der Anfrage enthalten sind. Der Anforderer zahlt jedoch nur für die Kosten der Anfrage. Der Bucket-Besitzer zahlt die Abrufgebühren.

Buckets mit Zahlung durch den Anforderer unterstützen die folgenden Funktionen nicht:

- Anonyme Anforderungen
- SOAPAnfragen
- Die Verwendung eines Buckets mit Zahlung durch den Anforderer als Ziel-Bucket für die Endbenutzer-Protokollierung, oder umgekehrt. Sie können jedoch die Endbenutzer-Protokollierung für einen Bucket mit Zahlung durch den Anforderer aktivieren, wenn es sich beim Ziel-Bucket nicht um einen Bucket mit Zahlung durch den Anforderer handelt.

Die Gebühren bei Zahlung durch den Anforderer

Die Verteilung der Gebühren für erfolgreiche Anforderungen mit Zahlung durch den Anforderer ist ganz einfach: Der Anforderer zahlt für die Datenübertragung und die Anforderung, der Bucket-Eigentümer für die Speicherung der Daten. Dem Bucket-Eigentümer werden jedoch nur unter den folgenden Bedingungen Gebühren für die Anforderung in Rechnung gestellt:

- Die Anfrage gibt einen Fehler `AccessDenied` (HTTP403 Forbidden) zurück und die Anfrage wird innerhalb der Person des Bucket-Besitzers initiiert AWS Konto oder AWS Organisation.
- Die Anfrage ist eine SOAP Anfrage.

Weitere Informationen zur Zahlung durch den Anforderer finden Sie in den folgenden Themen.

Themen

- [Konfigurieren von Zahlung durch den Anforderer für einen Bucket](#)
- [Die Konfiguration wird mit dem `requestPayment` abgerufen REST API](#)
- [Objekte aus Requester Pays-Buckets werden heruntergeladen](#)

Konfigurieren von Zahlung durch den Anforderer für einen Bucket

Sie können einen Amazon-S3-Bucket als Bucket mit Zahlung durch den Anforderer konfigurieren, damit anstelle des Bucket-Eigentümers der Anforderer die Kosten für die Anforderung und den Datendownload übernimmt.

Dieser Abschnitt enthält Beispiele für die Konfiguration von Requester Pays in einem Amazon S3 S3-Bucket mithilfe der Konsole und der RESTAPI.

Verwenden der S3-Konsole

So aktivieren Sie die Zahlung durch den Anforderer für einen S3-Bucket:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie die Zahlung durch den Anforderer aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).

4. Wählen Sie unter Requester pays (Zahlung durch Anforderer) die Option Edit (Bearbeiten).
5. Wählen Sie Enable (Aktivieren) und wählen Sie Save changes (Änderungen speichern) aus.

Amazon S3 aktiviert die Zahlung durch den Anforderer für Ihren Bucket und zeigt Ihre Bucket-Übersicht an. Unter Zahlung durch den Anforderer sehen Sie Aktiviert

Mit dem REST API

Nur der Bucket-Eigentümer kann den Konfigurationswert `RequestPaymentConfiguration.payer` eines Buckets auf `BucketOwner` (den Standardwert) oder `Requester` setzen. Die Einrichtung der Ressource `requestPayment` ist optional. Standardmäßig ist der Bucket kein Bucket mit Zahlung durch den Anforderer.

Um einen Bucket mit Zahlung durch den Anforderer in einen regulären Bucket umzuwandeln, verwenden Sie den Wert `BucketOwner`. In der Regel verwenden Sie `BucketOwner` zum Hochladen von Daten in den Amazon-S3-Bucket. Dann würden Sie den Wert auf `Requester` setzen, bevor Sie die Objekte in dem Bucket veröffentlichen.

Zum Einstellen requestPayment

- Verwenden Sie eine PUT-Anforderung, um den Wert `Payer` für einen bestimmten Bucket auf `Requester` zu setzen.

```
PUT ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Content-Length: 173
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

War die Anfrage erfolgreich, gibt Amazon S3 eine Antwort zurück, die etwa wie folgt aussieht.

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
```

```
Content-Length: 0
Connection: close
Server: AmazonS3
x-amz-request-charged:requester
```

Sie können Zahlung durch den Anforderer nur auf Bucket-Ebene festlegen. Sie können Zahlung durch den Anforderer nicht für bestimmte Objekte innerhalb des Buckets festlegen.

Sie können einen Bucket jederzeit als BucketOwner oder Requester konfigurieren. Es kann jedoch einige Minuten dauern, bis der neue Konfigurationswert aktiv ist.

Note

Besitzer von Buckets, die vorsigniert ausgeben, URLs sollten sorgfältig überlegen, bevor sie einen Bucket als Requester Pays konfigurieren, vor allem, wenn der Bucket eine lange Lebensdauer URL hat. Dem Bucket-Besitzer wird jedes Mal eine Gebühr in Rechnung gestellt, wenn der Anforderer einen vorsignierten Bucket verwendet, der URL die Anmeldeinformationen des Bucket-Besitzers verwendet.

Die Konfiguration wird mit dem requestPayment abgerufen REST API

Sie können den Payer-Wert ermitteln, der für einen Bucket eingerichtet ist, indem Sie die Ressource requestPayment abrufen.

Um die Ressource zurückzugeben requestPayment

- Verwenden Sie eine GET Anfrage, um die requestPayment Ressource abzurufen, wie in der folgenden Anfrage gezeigt.

```
GET ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

War die Anfrage erfolgreich, gibt Amazon S3 eine Antwort zurück, die etwa wie folgt aussieht.

```
HTTP/1.1 200 OK
```

```
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Type: [type]
Content-Length: [length]
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Diese Antwort zeigt, dass der `payer`-Wert auf `Requester` gesetzt ist.

Objekte aus Requester Pays-Buckets werden heruntergeladen

Den Anforderern werden die Gebühren für das Herunterladen von Daten aus Buckets mit Zahlung durch den Anforderer in Rechnung gestellt. Deshalb müssen die Anforderungen einen speziellen Parameter enthalten, `x-amz-request-payer`, der bestätigt, dass der Anforderer weiß, dass er Gebühren für den Download zahlen muss. Um auf Objekte in Buckets mit Zahlung durch den Anforderer zuzugreifen, müssen die Anforderungen eine der folgenden Informationen enthalten.

- Für `DELETE`, `GET`, und `PUT` Anfragen `HEADPOST`, fügen Sie sie `x-amz-request-payer : requester` in die Kopfzeile ein
- Wenn signierte URLs, fügen Sie es `x-amz-request-payer=requester` in die Anfrage ein

Ist die Anforderung erfolgreich und dem Auftraggeber werden Gebühren in Rechnung gestellt, enthält die Antwort den Header `x-amz-request-charged:requester`. Ist `x-amz-request-payer` nicht in der Anfrage enthalten, gibt Amazon S3 einen 403-Fehler zurück und stellt dem Bucket-Eigentümer die Gebühren für die Anfrage in Rechnung.

Note

Bucket-Eigentümer müssen ihren Anforderungen `x-amz-request-payer` nicht hinzufügen. Stellen Sie sicher, dass Sie `x-amz-request-payer` und seinen Wert in die Berechnung Ihrer Signatur aufgenommen haben. Weitere Informationen finden Sie unter [Konstruieren des CanonicalizedAmzHeaders Elements](#).

Mit dem REST API

Herunterladen von Objekten aus einem Bucket mit Zahlung durch den Anforderer

- Verwenden Sie eine GET-Anforderung, um ein Objekt aus einem Bucket mit Zahlung durch den Anforderer herunterzuladen, wie in der folgenden Anforderung gezeigt.

```
GET / [destinationObject] HTTP/1.1
Host: [BucketName].s3.amazonaws.com
x-amz-request-payer : requester
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Wenn die GET Anfrage erfolgreich ist und dem Anfragenden eine Gebühr berechnet wird, umfasst die Antwort Folgendes: `x-amz-request-charged:requester`

Amazon S3 kann für Anfragen, die versuchen, Objekte aus einem Bucket mit Zahlung durch den Anforderer abzurufen, einen `Access Denied`-Fehler zurückgeben. Weitere Informationen finden Sie unter [Fehlerantworten](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS CLI

Um Objekte aus einem Requester Pays-Bucket herunterzuladen, verwenden Sie AWS CLI, geben Sie `--request-payer requester` im Rahmen Ihrer `get-object` Anfrage an. Weitere Informationen finden Sie unter [get-object](#) im AWS CLI Referenz.

Bucket-Kontingente, Einschränkungen und Beschränkungen

Ein Amazon S3 S3-Bucket gehört dem AWS-Konto das hat ihn geschaffen. Die Bucket-Eigentümerschaft ist nicht auf ein anderes Konto übertragbar.

Bucket-Kontingentbeschränkungen

Standardmäßig können Sie in jedem Ihrer Buckets bis zu 100 Buckets erstellen AWS-Konten. Wenn Sie zusätzliche Buckets benötigen, können Sie das Kontingent Ihres Accounts auf maximal 1.000 Buckets erhöhen, indem Sie eine Anfrage zur Erhöhung des Kontingents einreichen. Es gibt keinen Leistungsunterschied, ungeachtet dessen, ob Sie viele Buckets oder nur wenige verwenden.

 Note

Sie müssen nicht für jeden Antrag auf eine Erhöhung des Kontingents mehrere Anträge einreichen. Ihr Bucket-Kontingents gilt für Ihre AWS-Region. Ihr Bucket-Kontingents gilt für Ihr AWS-Konto.

Informationen darüber, wie Sie Ihr Bucket-Kontingents erhöhen können, finden Sie unter [Amazon S3 S3-Endpunkte und Kontingents](#) in der Allgemeinen Amazon Web Services-Referenz.

Einschränkungen für Objekte und Buckets

Die maximale Bucket-Größe oder die Anzahl der Objekte, die Sie in einem Bucket speichern können, ist nicht begrenzt. Sie können alle Ihre Objekte in einem einzigen Bucket speichern, oder sie über mehrere Buckets verteilen. Sie können jedoch keinen Bucket über einen anderen Bucket erstellen.

Beschränkungen bei der Benennung von Buckets

Wenn Sie einen Bucket erstellen, wählen Sie dessen Namen und AWS-Region um ihn zu erstellen. Name oder Region einmal erstellter Buckets können nicht nachträglich geändert werden.

Wenn Sie einen Bucket benennen, sollten Sie einen Namen wählen, der für Sie oder Ihr Unternehmen relevant ist. Vermeiden Sie die Verwendung von Namen, die mit anderen Entitäten verbunden sind. Zum Beispiel sollten Sie es vermeiden, AWS oder Amazon in Ihrem Bucket-Namen zu verwenden.

Wiederverwenden von Bucket-Namen

Wenn ein Bucket leer ist, können Sie ihn löschen. Nachdem ein Bucket gelöscht wurde, wird der Name zur Wiederverwendung verfügbar. Nachdem Sie den Bucket gelöscht haben, können Sie den Namen aus verschiedenen Gründen möglicherweise jedoch nicht wiederverwenden.

Wenn Sie beispielsweise den Bucket löschen und der Name für die Wiederverwendung verfügbar wird, wird ein anderes AWS-Konto könnte einen Bucket mit diesem Namen erstellen. Außerdem kann einige Zeit vergehen, bis Sie den Namen eines gelöschten Buckets wiederverwenden können. Wenn Sie denselben Bucket-Namen verwenden möchten, empfehlen wir, den Bucket nicht zu löschen.

Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

Bucket-Benennung und automatisch erstellte Buckets

Wenn Ihre Anwendung automatisch Buckets erstellt, wählen Sie ein Bucket-Namensschema, das wahrscheinlich keine Namenskonflikte verursacht. Stellen Sie sicher, dass Ihre Anwendungslogik einen anderen Bucket-Namen auswählt, wenn ein Bucket-Name bereits vergeben ist.

Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

Bucket-Operationen

Das Hochverfügbarkeits-Engineering von Amazon S3 konzentriert sich auf get-, put-, list- und delete-Operationen. Da Bucket-Operationen mit einem zentralisierten, globalen Ressourcenbereich arbeiten, ist es nicht empfehlenswert, Buckets auf dem Hochverfügbarkeits-Codepfad Ihrer Anwendung zu erstellen, zu löschen oder zu konfigurieren. Sinnvoller ist es, Buckets in einer separaten Initialisierungs- oder Einrichtungsroutine zu erstellen, zu löschen oder zu konfigurieren, die seltener ausgeführt wird.

Hochladen, Herunterladen und Arbeiten mit Objekten in Amazon S3

Um Ihre Daten in Amazon S3 zu speichern, arbeiten Sie mit Ressourcen, die als Buckets und Objekte bezeichnet werden. Ein Bucket ist ein Container für Objekte. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben.

Um ein Objekt in Amazon S3 zu speichern, erstellen Sie einen Bucket und laden das Objekt dann in einen Bucket hoch. Wenn sich das Objekt im Bucket befindet, können Sie es öffnen, herunterladen und kopieren. Wenn Sie ein Objekt oder einen Bucket nicht mehr benötigen, können Sie diese Ressourcen aufräumen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Important

Wenn Sie in der Amazon S3 S3-Konsole für ein Objekt „Öffnen“ oder „Herunterladen als“ wählen, werden diese Operationen vorsigniertURLs. Für die Dauer von fünf Minuten ist Ihr Objekt für jeden zugänglich, der Zugriff auf diese URLs vorsignierten Objekte hat. Weitere Informationen zu Presigned finden Sie unter Presigned URLs [verwenden](#). URLs

Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon S3](#). Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter [AWS Kostenloses Kontingent](#).

Themen

- [Übersicht über Amazon-S3-Objekte](#)
- [Erstellen von Objektschlüsselnamen](#)

- [Arbeiten mit Objekt-Metadaten](#)
- [Objekte hochladen](#)
- [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#)
- [Fügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu](#)
- [Objekte kopieren, verschieben und umbenennen](#)
- [Herunterladen von Objekten](#)
- [Überprüfung der Objektintegrität](#)
- [Löschen von Amazon-S3-Objekten](#)
- [Organisieren, Auflisten und Arbeiten mit Ihren Objekten](#)
- [Sie arbeiten mit vorsignierten URLs](#)
- [Transformieren von Objekten mit S3 Object Lambda](#)

Übersicht über Amazon-S3-Objekte

Amazon S3 ist ein Objektspeicher, der eindeutige Schlüsselwerte verwendet, um so viele Objekte zu speichern, wie Sie möchten. Sie speichern diese Objekte in einem oder mehreren Buckets. Jedes Objekt kann eine Größe von bis zu 5 TB haben. Ein Objekt besteht aus Folgendem:

Schlüssel

Der Name, den Sie einem Objekt zuweisen. Zum Abrufen des Objekts verwenden Sie den Objektschlüssel. Weitere Informationen finden Sie unter [Arbeiten mit Objektmetadaten](#).

Versions-ID

In einem Bucket wird ein Objekt durch einen Schlüssel und eine Versions-ID eindeutig identifiziert. Die Versions-ID ist eine Zeichenfolge, die Amazon S3 generiert, wenn Sie das Objekt einem Bucket hinzufügen. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Value

Der Inhalt, den Sie speichern.

Ein Objektwert kann eine beliebige Bytefolge sein. Objekte können von 0 bis 5 TB groß sein. Weitere Informationen finden Sie unter [Objekte hochladen](#).

Metadaten

Ein Satz von Name-Wert-Paaren, mit dem Sie Informationen zum Objekt speichern können. Sie können Ihren Objekten in Amazon S3 Metadaten zuweisen, auch als benutzerdefinierte Metadaten bezeichnet. Amazon S3 weist diesen Objekten auch System-Metadaten hinzu, die es für die Verwaltung von Objekten verwendet. Weitere Informationen finden Sie unter [Arbeiten mit Objektmetadaten](#).

Subressourcen

Amazon S3 verwendet den Subressourcen-Mechanismus, um objektspezifische zusätzliche Informationen zu speichern. Subressourcen sind Objekten untergeordnet, deshalb sind werden immer einer anderen Entity zugeordnet, wie beispielsweise einem Objekt oder einem Bucket. Weitere Informationen finden Sie unter [Objekt-Subressourcen](#).

Informationen zur Zugriffskontrolle

Sie können den Zugriff auf Objekte kontrollieren, die Sie in Amazon S3 speichern. Amazon S3 unterstützt sowohl die ressourcenbasierte Zugriffskontrolle, wie z. B. eine Zugriffskontrollliste (ACL) und Bucket-Richtlinien, als auch die benutzerbasierte Zugriffskontrolle. Weitere Informationen über die Kontrolle des Zugriffs über finden Sie im:

- [Zugriffsverwaltung](#)
- [Identity and Access Management für Amazon S3](#)
- [Konfiguration ACLs](#)

Ihre Amazon-S3-Ressourcen (wie z. B. Buckets und Objekte) sind standardmäßig privat. Sie müssen anderen ausdrücklich eine Berechtigung erteilen, damit sie auf diese Ressourcen zugreifen dürfen. Weitere Informationen über das Teilen von Objekten finden Sie unter [Objekte mit vorsignierten teilen URLs](#).

Tags

Sie können Tags verwenden, um Ihre gespeicherten Objekte zu kategorisieren, für die Zugriffskontrolle oder die Kostenzuordnung. Weitere Informationen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

Objekt-Subressourcen

Amazon S3 definiert eine Menge von Subressourcen, die Buckets und Objekten zugeordnet sind. Subressourcen sind Objekten untergeordnet. Dies bedeutet, dass Subressourcen nicht alleine existieren. Sie sind immer einer anderen Entität zugeordnet, z. B. einem Objekt oder einem Bucket.

Die folgende Tabelle listet die Subressourcen auf, die Amazon-S3-Objekten zugeordnet sind.

Subressource	Beschreibung
acl	Enthält eine Liste der Rechte, wobei die Empfänger und die erteilten Berechtigungen identifiziert werden. Wenn Sie ein Objekt erstellen, identifiziert die <code>acl</code> den Objekteigentümer, der volle Kontrolle über das Objekt hat. Sie können ein Objekt abrufen ACL oder es durch eine aktualisierte Liste von Zuschüssen ersetzen. Bei jeder Aktualisierung eines ACL müssen Sie das vorhandene ersetzen ACL. Weitere Informationen zu finden ACLs Sie unter Übersicht über die Zugriffskontrollliste (ACL) .

Erstellen von Objektschlüsselnamen

Der Objektschlüssel (oder Schlüsselname) identifiziert das Objekt in einem Amazon-S3-Bucket eindeutig. Objekt-Metadaten bestehen aus Name/Wert-Paaren. Weitere Informationen zu Objekt-Metadaten erhalten Sie unter [Arbeiten mit Objekt-Metadaten](#).

Wenn Sie ein Objekt erstellen, geben Sie den Schlüsselnamen an, der das Objekt in dem Bucket eindeutig identifiziert. Wenn Sie in der [Amazon-S3-Konsole](#) einen Bucket markieren, wird beispielsweise eine Liste der Objekte in Ihrem Bucket angezeigt. Diese Namen sind die Objektschlüssel. Der Objektschlüsselname ist eine Folge von Unicode-Zeichen mit einer Kodierung von UTF -8 und einer Länge von bis zu 1.024 Byte. Bei den Objektschlüsselnamen muss die Groß- und Kleinschreibung beachtet werden.

Note

[Objektschlüsselnamen mit dem Wert „soap“ werden für virtual-hosted-style Anfragen nicht unterstützt.](#) Für Objektschlüsselnamenwerte, bei denen „soap“ verwendet wird, URL muss stattdessen ein [Pfadstil](#) verwendet werden.

Das Amazon-S3-Datenmodell ist eine flache Struktur: Sie erstellen einen Bucket und der Bucket speichert Objekte. Es gibt keine Hierarchie von Unterbuckets oder Unterordnern. Sie können jedoch mit den Schlüsselnamenpräfixen und Trennzeichen eine logische Hierarchie erschließen, wie in der Amazon-S3-Konsole. Die Amazon-S3-Konsole unterstützt ein Ordnerkonzept. Weitere Informationen zum Bearbeiten von Metadaten über die Amazon-S3-Konsole finden Sie unter [Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole](#).

Angenommen, Ihr Bucket (`admin-created`) enthält vier Objekte mit den folgenden Objektschlüsseln:

`Development/Projects.xls`

`Finance/statement1.pdf`

`Private/taxdocument.pdf`

`s3-dg.pdf`

Die Konsole verwendet die Schlüsselnamenpräfixe (`Development/`, `Finance/` und `Private/`) und das Trennzeichen ("`/`"), um eine Ordnerstruktur wie dargestellt anzuzeigen. Der `s3-dg.pdf`-Schlüssel hat kein Präfix, deshalb erscheint sein Objekt direkt auf Root-Ebene des Buckets. Wenn Sie den Ordner `Development/` öffnen, werden Sie darin das Objekt `Projects.xlsx` sehen.

- Amazon S3 unterstützt Buckets und Objekt. Es gibt keine Hierarchie in Amazon S3. Durch die Verwendung von Präfixen und Trennzeichen in einem Objektschlüsselnamen können jedoch die Amazon S3 S3-Konsole und AWS SDKskann Hierarchien ableiten und das Konzept von Ordnern einführen.
- Die Amazon-S3-Konsole implementiert die Ordnerobjekterstellung, indem sie ein Null-Byte-Objekt mit dem Wert des Ordners für Präfix und Trennzeichen als Schlüssel erstellt. Diese Ordnerobjekte werden nicht in der Konsole angezeigt. Ansonsten verhalten sie sich wie alle anderen Objekte und können mit Hilfe von, betrachtet und manipuliert werden REST API AWS CLI, und AWS SDKs.

Richtlinien für Objektschlüsselnamen

Sie können ein beliebiges UTF -8-Zeichen in einem Objektschlüsselnamen verwenden. Die Verwendung bestimmter Zeichen in Schlüsselnamen kann jedoch bei manchen Anwendungen und Protokollen zu Problemen führen. Die folgenden Richtlinien helfen Ihnen dabei, die Einhaltung von websicheren ZeichenDNS, XML Parsern und anderen zu maximieren. APIs

Sichere Zeichen

Die folgenden Zeichensätze sind allgemein sicher für die Verwendung in Schlüsselnamen.

- | | |
|-------------------------|--|
| Alphanumerische Zeichen | <ul style="list-style-type: none">• 0-9• a-z• A-Z |
| Sonderzeichen | <ul style="list-style-type: none">• Ausrufezeichen (!)• Bindestrich (-)• Unterstrich (_)• Punkt (.)• Sternchen (*)• Einzelnes Anführungszeichen (')• Öffnende Klammer ((• Schließende Klammer ()) |

Nachfolgend finden Sie Beispiele für gültige Objektschlüsselnamen:

- 4my-organization
- my.great_photos-2014/jan/myvacation.jpg
- videos/2014/birthday/video1.wmv

Note

Bei Objekten mit Schlüsselnamen, die mit Punkt(en) "." enden, die über die Amazon-S3-Konsole heruntergeladen werden, werden sämtliche Punkte "." aus dem Schlüsselnamen des heruntergeladenen Objekts entfernt. Um ein Objekt herunterzuladen, dessen Schlüsselname auf Punkt (en) „.“ endet und im heruntergeladenen Objekt beibehalten wird, müssen Sie die AWS Command Line Interface (AWS CLI), AWS SDKs, oder RESTAPI.

Darüber hinaus sollten Sie folgende Prefixbeschränkungen beachten:

- Objekte mit dem Prefix „./“ müssen mit dem AWS Command Line Interface (AWS CLI), AWS SDKs, oder RESTAPI. Sie können die Amazon-S3-Konsole nicht verwenden.

- Objekte mit dem Präfix „../“ können nicht hochgeladen werden AWS Command Line Interface (AWS CLI) oder Amazon S3 S3-Konsole.

Zeichen, die möglicherweise eine Sonderverarbeitung benötigen

Die folgenden Zeichen in einem Schlüsselnamen erfordern möglicherweise zusätzliche Codebehandlung und müssen wahrscheinlich URL codiert oder als HEX referenziert werden. Einige davon sind nicht darstellbare Zeichen, und Ihr Browser kann sie ggf. nicht verarbeiten, was zudem einer speziellen Vorgehensweise bedarf:

- Ampersand ("&")
- Dollar ("\$")
- ASCII-Zeichenbereiche 00—1F hex (0—31 dezimal) und 7F (127 dezimal)
- 'At'-Symbol ("@")
- Gleichheitszeichen ("=")
- Semikolon (";")
- Schrägstrich ("/")
- Doppelpunkt (":")
- Plus ("+")
- Leerzeichen – Wichtige Leerzeichenfolgen gehen möglicherweise bei bestimmten Verwendungszwecken verloren (insbesondere Mehrfachleerzeichen).
- Komma (",")
- Fragezeichen ("?")

Zeichen, die Sie vermeiden sollten

Wir empfehlen, die folgenden Zeichen nicht in einem Schlüsselnamen zu verwenden, da die Behandlung von Sonderzeichen häufig vorkommt und diese nicht in allen Anwendungen einheitlich ist.

- Umgekehrter Schrägstrich ("\")
- Linke geschweifte Klammer ("{"

- Nicht druckbare ASCII Zeichen (128—255 Dezimalzeichen)
- Caret ("^")
- Rechte geschweifte Klammer ("}")
- Prozentzeichen ("%")
- Accent Grave ("`")
- Rechte eckige Klammer ("]")
- Anführungszeichen
- Größersymbol (">")
- Linke eckige Klammer ("[")
- Tilde ("~")
- Kleiner als-Zeichen („<“)
- Pfundzeichen ("£")
- Vertikaler Strich ("|")

XMLSchlüsseleinschränkungen für verwandte Objekte

Gemäß der [XMLNorm zur end-of-line Verarbeitung](#) ist der gesamte XML Text normalisiert, sodass Zeilenumbrüche (ASCII Code 13) und Zeilenumbrüche, denen unmittelbar ein Zeilenvorschub folgt (ASCII Code 10), durch ein einzelnes Zeilenvorschubzeichen ersetzt werden. Um die korrekte Analyse von Objektschlüsseln in XML Anfragen zu gewährleisten, [müssen Zeilenumbrüche und andere Sonderzeichen durch ihren entsprechenden XML Entitätscode ersetzt werden](#), wenn sie in XML Tags eingefügt werden. Im Folgenden finden Sie eine Liste solcher Sonderzeichen und ihrer entsprechenden Entitätscodes:

- ' wie '
- " wie "
- & wie &
- < wie <
- > wie >
- \r wie  oder 
- \n wie
 oder

Example

Das folgende Beispiel veranschaulicht die Verwendung eines XML Entitätscodes als Ersatz für einen Zeilenumbruch. Diese DeleteObjects-Anforderung löscht ein Objekt mit dem key-Parameter: /some/prefix/objectwith\r\n (wobei \r die Zeilenumschaltung ist).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Object>
    <Key>/some/prefix/objectwith\r\n</Key>
  </Object>
</Delete>
```

Arbeiten mit Objekt-Metadaten

Sie können Objekt-Metadaten in Amazon S3 zum Zeitpunkt des Hochladens des Objekts festlegen. Objekt-Metadaten bestehen aus Name/Wert-Paaren. Nachdem Sie das Objekt hochgeladen haben, können Sie Objekt-Metadaten nicht mehr ändern. Die einzige Methode, wie Sie Objekt-Metadaten ändern können, ist es, eine Kopie des Objekts anzulegen und die Metadaten festzulegen.

Wenn Sie ein Objekt erstellen, geben Sie auch den Schlüsselnamen an, der das Objekt in dem Bucket eindeutig identifiziert. Der Objektschlüssel (oder Schlüsselname) identifiziert das Objekt in einem Amazon-S3-Bucket eindeutig. Weitere Informationen finden Sie unter [Erstellen von Objektschlüsselnamen](#).

Es gibt zweierlei Arten von Metadaten in Amazon S3: systemdefinierte und benutzerdefinierte Metadaten. In den folgenden Abschnitten finden Sie weitere Informationen zu systemdefinierten und benutzerdefinierten Metadaten. Weitere Informationen zum Bearbeiten von Metadaten mit der Amazon-S3-Konsole finden Sie unter [Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole](#).

Systemdefinierte Objektmetadaten

Amazon S3 verwaltet für jedes in einem Bucket gespeicherte Objekt einen Satz Systemmetadaten. Amazon S3 verarbeitet diese Systemmetadaten nach Bedarf. Beispielsweise verwaltet Amazon S3 das Erstellungsdatum eines Objekts und seine Größe als Metadaten und verwendet dies Information für die Objektverwaltung.

Es gibt zwei Kategorien von Systemmetadaten:

- Systemdefiniert – Metadaten wie beispielsweise das Erstellungsdatum eines Objekts werden vom System definiert. Dies bedeutet, dass nur Amazon S3 den Wert ändern kann.

- Benutzerdefiniert – Andere Systemmetadaten, wie beispielsweise die für das Objekt konfigurierte Speicherklasse und die Angabe, ob für das Objekt eine serverseitige Verschlüsselung aktiviert ist, sind Beispiele für Metadaten, deren Werte Sie definieren können. Wenn Ihr Bucket als Website konfiguriert ist, möchten Sie manchmal eine Seitenanfrage auf eine andere oder externe URL Seite umleiten. In diesem Fall ist eine Webseite ein Objekt in Ihrem Bucket. Amazon S3 speichert den Wert für die Seitenumleitung als Systemmetadaten, deren Wert Sie steuern können.

Wenn Sie Objekte erstellen, können Sie die Werte dieser Systemmetadaten konfigurieren oder sie nach Bedarf aktualisieren. Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Amazon S3 verwendet AWS KMS Schlüssel zum Verschlüsseln Ihrer Amazon S3 S3-Objekte. AWS KMS verschlüsselt nur die Objektdaten. Die Prüfsumme wird zusammen mit dem angegebenen Algorithmus als Teil der Metadaten des Objekts gespeichert. Wenn für das Objekt eine serverseitige Verschlüsselung angefordert wird, wird die Prüfsumme in verschlüsselter Form gespeichert. Weitere Informationen zur serverseitigen Verschlüsselung finden Sie unter [Datenschutz durch Verschlüsselung](#).

Note

Der PUT-Anforderungs-Header ist auf eine Größe von 8 KB begrenzt. Innerhalb des PUT-Anforderungs-Headers sind die systemdefinierten Metadaten auf eine Größe von 2 KB begrenzt. Die Größe systemdefinierter Metadaten wird anhand der Summe der Anzahl der Byte in der ASCII US-Kodierung jedes Schlüssels und Werts gemessen.

Die folgenden Tabelle enthält eine Liste der vom System definierten Metadaten, und gibt an, ob Sie sie aktualisieren können.

Name	Beschreibung	Kann der Benutzer den Wert ändern?
Date	Aktuelles Datum und Uhrzeit	Nein

Name	Beschreibung	Kann der Benutzer den Wert ändern?
Cache-Control	Ein allgemeines Header-Feld zum Angeben von Caching-Richtlinien.	Ja
Content-Disposition	Darstellungsinformationen zum Objekt.	Ja
Content-Length	Die Objektgröße in Bytes	Nein
Content-Type	Der Typ des Objekts.	Ja
Last-Modified	Das Datum, an dem das Objekt erstellt wurde, oder das letzte Änderungsdatum, je nachdem, welcher Wert aktueller ist. Bei mehrteiligen Uploads entspricht das Objekterstellungsdatum dem Startdatum des mehrteiligen Uploads.	Nein
ETag	Ein Entity-Tag (ETag), das eine bestimmte Version eines Objekts darstellt. Bei Objekten, die nicht als mehrteiliger Upload hochgeladen werden und entweder unverschlüsselt oder durch serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt sind, ETag handelt es sich um eine MD5 Zusammenfassung der Daten.	Nein
x-amz-server-side-encryption	Ein Header, der angibt, ob die serverseitige Verschlüsselung für das Objekt aktiviert ist und ob diese Verschlüsselung die AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder die Verwendung von verwalteten Amazon S3 S3-Verschlüsselungsschlüsseln (SSE-S3). Weitere Informationen finden Sie unter Schützen von Daten mit serverseitiger Verschlüsselung .	Ja

Name	Beschreibung	Kann der Benutzer den Wert ändern?
<code>x-amz-checksum-crc32</code> , <code>x-amz-checksum-crc32c</code> , <code>x-amz-checksum-sha1</code> , <code>x-amz-checksum-sha256</code>	Header, die die Prüfsumme oder den Digest des Objekts enthalten. Je nach Prüfsummenalgorithmus, den Amazon S3 verwenden soll, wird höchstens jeweils einer dieser Header festgelegt. Weitere Informationen zur Auswahl des Prüfsummenalgorithmus finden Sie unter Überprüfung der Objektintegrität .	Nein
<code>x-amz-version-id</code>	Die Objekt-Versionsverwaltung. Wenn Sie die Versionsverwaltung für einen Bucket aktivieren, weist Amazon S3 allen Objekten, die dem Bucket hinzugefügt werden, eine Versions-ID zu. Weitere Informationen finden Sie unter Verwenden der Versioning in S3-Buckets .	Nein
<code>x-amz-delete-marker</code>	Ein boolesches Kennzeichen, das angibt, ob das Objekt eine Löschmarkierung ist. Dieses Kennzeichen wird nur in Buckets verwendet, für die die Versionsverwaltung aktiviert ist.	Nein
<code>x-amz-storage-class</code>	Die Speicherklasse, für die Speicherung des Objekts verwendet wird. Weitere Informationen finden Sie unter Verwenden von Amazon-S3-Speicherklassen .	Ja
<code>x-amz-website-redirect-location</code>	Ein Header, der Anfragen für das zugehörige Objekt an ein anderes Objekt im selben Bucket oder an ein externes URL Objekt weiterleitet. Weitere Informationen finden Sie unter (Optional) Konfigurieren einer Webseitenumleitung .	Ja

Name	Beschreibung	Kann der Benutzer den Wert ändern?
<code>x-amz-server-side-encryption-aws-kms-key-id</code>	Ein Header, der die ID des AWS KMS symmetrischer KMS Verschlüsselungsschlüssel, der zum Verschlüsseln des Objekts verwendet wurde. Dieser Header wird nur verwendet, wenn der Header <code>x-amz-server-side-encryption</code> vorhanden ist und den Wert <code>aws:kms</code> aufweist.	Ja
<code>x-amz-server-side-encryption-customer-algorithm</code>	Ein Header, der angibt, ob die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) aktiviert ist. Weitere Informationen finden Sie unter Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE .	Ja
<code>x-amz-tagging</code>	Das Tag-Set für das Objekt. Das Tag-Set muss als Abfrageparameter codiert werden. URL	Ja

Benutzerdefinierte Objektmetadaten

Wenn Sie ein Objekt hochladen, können Sie ihm Metadaten zuweisen. Sie geben diese optionalen Informationen als Name-Wert-Paar (Schlüssel-Wert) an, wenn Sie eine PUT- oder POST-Anforderung senden, um das Objekt zu erstellen. Wenn Sie Objekte mithilfe von hochladen RESTAPI, müssen die optionalen benutzerdefinierten Metadatenamen mit `beginnen`, um sie von anderen `x-amz-meta-` Headern zu unterscheiden. HTTP Wenn Sie das Objekt mithilfe von abrufen RESTAPI, wird dieses Präfix zurückgegeben. Wenn Sie Objekte mit dem hochladen SOAPAPI, ist das Präfix nicht erforderlich. Wenn Sie das Objekt mithilfe von abrufen SOAPAPI, wird das Präfix entfernt, unabhängig davon, welches API Sie zum Hochladen des Objekts verwendet haben.

Note

SOAP Support Over HTTP ist veraltet, SOAP ist aber weiterhin über verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstützt SOAP. Anstatt zu verwenden SOAP, empfehlen wir, entweder den REST API oder den AWS SDKs.

Wenn Metadaten über die abgerufen werden REST API, fasst Amazon S3 Header mit demselben Namen (ohne Berücksichtigung der Groß- und Kleinschreibung) zu einer kommagetrennten Liste zusammen. Wenn einige Metadaten nicht darstellbare Zeichen enthalten, werden sie nicht zurückgegeben. Stattdessen wird der `x-amz-missing-meta`-Header mit der Anzahl der nicht darstellbaren Metadateneinträge als Wert zurückgegeben. Die `HeadObject`-Aktion ruft Metadaten von einem Objekt ab, ohne das Objekt selbst zurückzugeben. Dieser Vorgang ist nützlich, wenn Sie nur an den Metadaten eines Objekts interessiert sind. Um HEAD verwenden zu können, müssen Sie READ-Zugriff auf das Objekt haben. Weitere Informationen finden Sie [HeadObject](#) in der Amazon Simple Storage Service API Reference.

Benutzerdefinierte Metadaten sind Schlüssel/Wert-Paare. Amazon S3 speichert benutzerdefinierte Metadaten in Kleinbuchstaben.

Amazon S3 erlaubt beliebige Unicode-Zeichen in Ihren Metadatenwerten.

Um Probleme bei der Darstellung dieser Metadatenwerte zu vermeiden, sollten Sie bei der Verwendung von ASCII US-Zeichen REST und bei der Verwendung von -8 bei der Verwendung von SOAP oder browserbasierten Uploads über die Zeichen UTF -8 einhalten. POST

Wenn Sie in Ihren Metadatenwerten ASCII Nicht-US-Zeichen verwenden, wird die bereitgestellte Unicode-Zeichenfolge auf Nicht-US-Zeichen untersucht. ASCII [Die Werte solcher Header werden vor dem Speichern gemäß 2047 dekodiert und gemäß RFC2047 codiert, um sie vor der Rückgabe mailsicher zu machen. RFC](#) Wenn die Zeichenfolge nur ASCII US-Zeichen enthält, wird sie unverändert dargestellt.

Im Folgenden wird ein Beispiel gezeigt.

```
PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÖÑ S3
```

```
HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=  
  
PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3  
  
HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3
```

Note

Der PUT-Anforderungs-Header ist auf eine Größe von 8 KB begrenzt. Innerhalb des PUT-Anforderungs-Headers sind die benutzerdefinierten Metadaten auf eine Größe von 2 KB begrenzt. Die Größe benutzerdefinierter Metadaten wird anhand der Summe der Byteanzahl in der UTF -8-Kodierung jedes Schlüssels und Werts gemessen.

Informationen zum Ändern der Metadaten Ihres Objekts, nachdem das Objekt hochgeladen wurde, indem eine Kopie des Objekts erstellt und geändert und das alte Objekt ersetzt wurde oder indem eine neue Version erstellt wurde, finden Sie unter [Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole](#).

Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole

Sie können die Amazon-S3-Konsole verwenden, um Metadaten von vorhandenen S3-Objekten zu bearbeiten. Einige Metadaten werden von Amazon S3 festgelegt, wenn Sie das Objekt hochladen. Beispielsweise sind Content-Length und Last-Modified systemdefinierte Objektmetadatenfelder, die von einem Benutzer nicht geändert werden können.

Sie können einige Metadaten festlegen, wenn Sie das Objekt hochladen. Es kann danach je nach Anforderungen bearbeitet werden. Beispielsweise haben Sie vielleicht eine Reihe von Objekten, die Sie ursprünglich in der STANDARD-Speicherklasse gespeichert haben. Im Laufe der Zeit benötigen Sie diese Daten möglicherweise nicht mehr, um hochverfügbar zu sein. Sie können also die Speicherklasse in GLACIER ändern, indem Sie den Wert des x-amz-storage-class-Schlüssels von STANDARD nach GLACIER bearbeiten.


 Note

Berücksichtigen Sie die folgenden Probleme, wenn Sie Objektmetadaten in Amazon S3 bearbeiten:

- Dieser Vorgang erstellt eine Kopie des Objekts mit aktualisierten Einstellungen und dem Datum der letzten Änderung. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Wenn das S3-Versioning nicht aktiviert ist, ersetzt eine neue Kopie des Objekts das ursprüngliche Objekt. Das Tool AWS-Konto Der IAM Rolle zugeordnet, die die Eigenschaft ändert, wird auch der Besitzer des neuen Objekts oder (Objektversion).
- Um die Amazon S3 S3-Konsole zum Bearbeiten von Metadaten für ein Objekt mit benutzerdefinierten Tags zu verwenden, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Berechtigung. Wenn Sie die Amazon S3 S3-Konsole verwenden, um die Metadaten für ein Objekt zu bearbeiten, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Genehmigung.

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, werden die Metadaten für das Objekt aktualisiert, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

- Das Bearbeiten von Metadaten aktualisiert die Werte für vorhandene Schlüsselnamen.
- Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mit der Konsole kopiert werden. Sie müssen das verwenden AWS CLI, AWS SDK, oder der Amazon S3 RESTAPI.

 Warning

Warten Sie beim Bearbeiten von Metadaten von Ordnern, bis der Vorgang `Edit metadata` abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte bearbeitet werden.

In den folgenden Themen wird beschrieben, wie Metadaten eines Objekts mithilfe der Amazon-S3-Konsole bearbeitet werden.

Bearbeiten von systemdefinierten Metadaten

Sie können verschiedene jedoch nicht alle System-Metadaten für ein S3-Objekt konfigurieren. Eine Liste der systemdefinierten Metadaten und ob Sie ihre Werte ändern können, finden Sie unter [Systemdefinierte Objektmetadaten](#).

Bearbeiten systemdefinierter Metadaten eines Objekts

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Navigieren Sie zu Ihrem Amazon-S3-Bucket oder -Ordner und aktivieren Sie das Kontrollkästchen links neben den Namen der Objekte mit Metadaten, die Sie bearbeiten möchten.
3. Klicken Sie im Menü Actions (Aktionen) auf Edit actions (Aktionen bearbeiten) und wählen Sie Edit metadata (Metadaten bearbeiten).
4. Überprüfen Sie die aufgeführten Objekte und wählen Sie Add metadata (Metadaten hinzufügen).
5. Wählen Sie für Type(Typ) der Metadaten, System-defined (systemdefiniert) aus.
6. Geben Sie einen eindeutigen Key (Schlüssel) und den Value (Wert) der Metadaten an.
7. Um weitere Metadaten zu bearbeiten, wählen Sie Add metadata (Metadaten hinzufügen). Sie können auch Entfernen wählen, um einen Satz von zu entfernen type-key-values.
8. Wenn Sie fertig sind, wählen Sie Edit metadata (Metadaten ändern). Amazon S3 ändert daraufhin die Metadaten der angegebenen Objekte.

Bearbeiten von benutzerdefinierten Metadaten

Sie können benutzerdefinierte Metadaten eines Objekts bearbeiten, indem Sie das Metadatenpräfix `x-amz-meta-` und einen Namen miteinander kombinieren, um einen benutzerdefinierten Schlüssel zu generieren. Wenn Sie beispielsweise den benutzerdefinierten Namen `alt-name` hinzufügen, wäre der Metadaten-Schlüssel `x-amz-meta-alt-name`.

Benutzerdefinierte Metadaten können insgesamt bis zu 2 KB umfassen. Um die Gesamtgröße der benutzerdefinierten Metadaten zu berechnen, addieren Sie die Anzahl der Byte in der UTF-8-Kodierung für jeden Schlüssel und Wert. Beide Schlüssel und ihre Werte müssen den US-Standards entsprechen. ASCII Weitere Informationen finden Sie unter [Benutzerdefinierte Objektmetadaten](#).

So bearbeiten Sie benutzerdefinierte Metadaten eines Objekts

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, denen Sie Metadaten hinzufügen möchten.

Sie können auch optional zu einem Ordner navigieren.

3. Aktivieren Sie in der Liste Objects (Objekte) das Kontrollkästchen neben den Namen der Objekte, denen Sie Metadaten hinzufügen möchten.
4. Wählen Sie im Menü Actions (Aktionen) die Option Edit metadata (Metadaten bearbeiten).
5. Überprüfen Sie die aufgeführten Objekte und wählen Sie Add metadata (Metadaten hinzufügen).
6. Wählen Sie für Type (Typ) der Metadaten, User-defined (benutzerdefiniert) aus.
7. Geben Sie einen auf x-amz-meta- folgenden eindeutigen, benutzerdefinierten Key (Schlüssel) ein. Geben Sie ebenfalls einen Value (Wert) der Metadaten ein.
8. Um zusätzliche Metadaten hinzuzufügen, wählen Sie Add metadata (Metadaten hinzufügen). Sie können auch Entfernen wählen, um einen Satz von zu entfernen type-key-values.
9. Wählen Sie Edit metadata (Metadaten bearbeiten).

Amazon S3 bearbeitet die Metadaten der angegebenen Objekte.

Objekte hochladen

Wenn Sie eine Datei Amazon S3 hochladen, wird sie als S3-Objekt gespeichert. Objekte umfassen die Datei und die Metadaten, die das Objekt beschreiben. Ein Bucket kann eine unbegrenzte Anzahl von Objekten aufnehmen. Bevor Sie Dateien zu Amazon S3 hochladen können, benötigen Sie die Schreibberechtigungen für den Bucket. Weitere Informationen zu den Zugriffsberechtigungen finden Sie unter [Identity and Access Management für Amazon S3](#).

Sie können beliebige Dateitypen – Bilder, Backups, Daten, Filme usw. – in einen S3-Bucket hochladen. Die maximale Größe einer Datei, die Sie über die Amazon-S3-Konsole hochladen können, ist 160 GB. Um eine Datei hochzuladen, die größer als 160 GB ist, verwenden Sie AWS Command Line Interface (AWS CLI), AWS SDKs oder Amazon S3 RESTAPI.

Wenn Sie ein Objekt mit einem Schlüsselnamen hochladen, der bereits in einem Bucket mit aktiviertem Versioning vorhanden ist, erstellt Amazon S3 eine weitere Version des Objekts, statt das

vorhandene Objekt zu ersetzen. Weitere Informationen zur Aktivierung der Versionierung finden Sie unter [Aktivieren des Versioning für Buckets](#).

Abhängig von der Größe der Daten, die Sie hochladen, bietet Amazon S3 die folgenden Optionen:

- Laden Sie ein Objekt in einem einzigen Vorgang hoch, indem Sie AWS SDKs, RESTAPI, oder AWS CLI— Mit einem einzigen PUT Vorgang können Sie ein einzelnes Objekt mit einer Größe von bis zu 5 GB hochladen.
- Hochladen eines einzelnen Objekts mit der Amazon-S3-Konsole – Mit der Amazon-S3-Konsole können Sie ein einzelnes Objekt mit einer Größe von bis zu 160 GB hochladen.
- Laden Sie ein Objekt in Teilen hoch, indem Sie AWS SDKs, RESTAPI, oder AWS CLI— Mithilfe des mehrteiligen API Upload-Vorgangs können Sie ein einzelnes großes Objekt mit einer Größe von bis zu 5 TB hochladen.

Der mehrteilige API Upload-Vorgang wurde entwickelt, um das Hochladen größerer Objekte zu verbessern. Sie können ein Objekt in Teilen hochladen. Diese Objektteile können unabhängig, in jeder beliebigen Reihenfolge und parallel hochgeladen werden. Sie können einen mehrteiligen Upload verwenden, um Objekte mit einer Größe von 5 MB bis 5 TB hochzuladen. Weitere Informationen finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Wenn Sie ein Objekt hochladen, wird das Objekt standardmäßig automatisch mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Wenn Sie es herunterladen, wird das Objekt entschlüsselt. Weitere Informationen erhalten Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#) und [Datenschutz durch Verschlüsselung](#).

Wenn Sie ein Objekt hochladen und eine andere Art der Standardverschlüsselung verwenden möchten, können Sie auch die serverseitige Verschlüsselung mit angeben AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) in Ihren PUT S3-Anfragen oder legen Sie die Standardverschlüsselungskonfiguration im Ziel-Bucket so fest, dass SSE - KMS zur Verschlüsselung Ihrer Daten verwendet werden soll. Weitere Hinweise zu SSE - finden Sie KMS unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#). Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, müssen Sie über die entsprechende Berechtigung verfügen. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [Erstellen von KMS Schlüsseln, die andere Konten verwenden können](#), in [AWS Key Management Service Leitfaden für Entwickler](#).

Wenn Sie in Amazon S3 auf den Fehler Access Denied (403 Forbidden) stoßen, finden Sie weitere Informationen [Problembehandlung bei Zugriff verweigert \(403 Forbidden\) Fehler in Amazon S3](#) zu den häufigsten Ursachen unter.

Hochladen eines Objekts

Verwenden der S3-Konsole

In diesem Verfahren wird erläutert, wie Sie Objekte und Ordner mithilfe der Konsole in einen Amazon-S3-Bucket hochladen.

Wenn Sie ein Objekt hochladen, ist der Name des Objektschlüssels der Dateiname und etwaige optionale Präfixe. In der Amazon-S3-Konsole können Sie Ordner erstellen, um Ihre Objekte zu organisieren. In Amazon S3 werden Ordner als Präfixe dargestellt, die im Namen des Objektschlüssels angezeigt werden. Wenn Sie ein einzelnes Objekt in einen Ordner in der Amazon-S3-Konsole hochladen, wird der Ordnername im Namen des Objektschlüssels aufgenommen.

Wenn Sie beispielsweise ein Objekt mit dem Namen `sample1.jpg` in einen Ordner namens `backup` hochladen, lautet der Schlüsselname `backup/sample1.jpg`. Allerdings wird das Objekt in der Konsole als `sample1.jpg` im Ordner `backup` angezeigt. Weitere Informationen zu Schlüsselnamen finden Sie unter [Arbeiten mit Objekt-Metadaten](#).

Note

Wenn Sie ein Objekt umbenennen oder eine der Eigenschaften in der Amazon-S3-Konsole ändern, z. B. Speicherklasse, Verschlüsselung oder Metadaten, wird ein neues Objekt erstellt, das das alte ersetzt. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).

Wenn Sie einen Ordner hochladen, lädt Amazon S3 alle Dateien und Unterordner aus dem angegebenen Ordner in Ihren Bucket hoch. Es wird dann ein Objektschlüsselname zugewiesen, der eine Kombination aus dem hochgeladenen Dateinamen und dem Ordnernamen ist. Wenn Sie beispielweise einen Ordner mit dem Namen `/images` hochladen, der zwei Dateien, `sample1.jpg` und `sample2.jpg`, enthält, lädt Amazon S3 die Dateien hoch und weist ihnen die jeweiligen Schlüsselnamen `images/sample1.jpg` und `images/sample2.jpg` zu. Der Schlüsselname enthält den Ordnernamen als Präfix. Die Amazon-S3-Konsole zeigt nur den Teil

des Schlüsselnamens nach dem letzten / an. Beispielsweise werden in einem images-Ordner die Objekte images/sample1.jpg und images/sample2.jpg als sample1.jpg und sample2.jpg angezeigt.

So laden Sie Ordner und Dateien in einen S3-Bucket hoch

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in den Ihre Ordner oder Dateien hochgeladen werden sollen.
4. Klicken Sie auf Upload.
5. Führen Sie im Fenster Upload einen der folgenden Schritte aus:
 - Ziehen Sie Dateien und Ordner in das Fenster Upload (Hochladen) und legen Sie sie dort ab.
 - Wählen Sie Datei hinzufügen oder Ordner hinzufügen aus, wählen Sie die hochzuladenden Dateien oder Ordner und anschließend Öffnen aus.
6. Um die Versioning zu aktivieren, wählen Sie unter Destination (Ziel) die Option Enable Bucket Versioning (Bucket-Versioning aktivieren) aus.
7. Um die aufgelisteten Dateien und Ordner hochzuladen, ohne zusätzliche Upload-Optionen zu konfigurieren, wählen Sie unten auf der Seite Upload aus.

Amazon S3 lädt Ihre Objekte und Ordner hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: Status eine Erfolgsmeldung angezeigt.

So konfigurieren Sie zusätzliche Objekteigenschaften

1. Um Zugriffssteuerungslisten-Berechtigungen zu ändern, wählen Sie Permissions (Berechtigungen) aus.
2. Bearbeiten Sie unter Zugriffskontrollliste (ACL) die Berechtigungen.

Weitere Informationen zu den Zugriffsberechtigungen für Objekte finden Sie unter [Verwenden Sie die S3-Konsole, um ACL Berechtigungen für ein Objekt festzulegen](#). Sie können der Öffentlichkeit Lesezugriff auf Ihre Objekte erteilen, und zwar für alle Dateien, die Sie hochladen. Wir empfehlen allerdings, die Standardeinstellung für den öffentlichen Lesezugriff nicht zu ändern. Die Erteilung von öffentlichem Lesezugriff ist nur für wenige Anwendungsfälle sinnvoll, beispielsweise, wenn Buckets für Websites verwendet werden. Nachdem Sie das

Objekt hochgeladen haben, können Sie jederzeit Änderungen an den Objektberechtigungen vornehmen.

3. Um andere zusätzliche Eigenschaften zu konfigurieren, wählen Sie Properties (Eigenschaften) aus.
4. Wählen Sie im Abschnitt Speicherklasse die Speicherklasse für die Dateien aus, die Sie hochladen.

Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

5. Um die Verschlüsselungseinstellungen für Ihre Objekte zu aktualisieren, gehen Sie unter Server-side encryption settings (Serverseitige Verschlüsselungseinstellungen) wie folgt vor.
 - a. Klicken Sie auf Specify an encryption key (Verschlüsselungsschlüssel angeben).
 - b. Wählen Sie unter Verschlüsselungseinstellungen die Option Verwenden von Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung aus.
 - c. Wenn Sie Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung ausgewählt haben, müssen Sie die folgenden Verschlüsselungseinstellungen konfigurieren.
 - Um die hochgeladenen Dateien mit Schlüsseln zu verschlüsseln, die von Amazon S3 verwaltet werden, wählen Sie Amazon S3 managed key (SSE-S3).

Weitere Informationen finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

- Um die hochgeladenen Dateien mithilfe von Schlüsseln zu verschlüsseln, die in gespeichert sind AWS Key Management Service (AWS KMS), wählen Sie AWS Key Management Service Schlüssel (SSE-KMS). Wählen Sie dann eine der folgenden Optionen für AWS KMS Schlüssel:
 - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

- Um den KMS Schlüssel einzugebenARN, wählen Sie **Enter AWS KMS key ARN**, und geben Sie dann Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie **Create a KMS Key** aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

 **Important**

Sie können nur KMS Schlüssel verwenden, die in derselben Version verfügbar sind AWS-Region wie der Eimer. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN.

Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter [Identifizieren symmetrischer und KMS asymmetrischer Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

6. Wählen Sie **On (Ein)** aus, um zusätzliche Prüfsummen zu verwenden. Wählen Sie dann für **Checksum function (Prüfsummenfunktion)** die Funktion aus, die Sie verwenden möchten. Amazon S3 berechnet und speichert den Prüfsummenwert, nachdem es das gesamte Objekt erhalten hat. Sie können das Feld **Precalculated value (Vorberechneter Wert)** verwenden, um einen vorberechneten Wert anzugeben. In diesem Fall vergleicht Amazon S3 den von Ihnen angegebenen Wert mit dem eigenen berechneten Wert. Wenn die beiden Werte nicht übereinstimmen, generiert Amazon S3 einen Fehler.

Mit zusätzlichen Prüfsummen können Sie den Prüfsummenalgorithmus angeben, den Sie zur Überprüfung Ihrer Daten verwenden möchten. Weitere Informationen zu zusätzlichen Prüfsummen finden Sie unter [Überprüfung der Objektintegrität](#).

7. Um allen hochgeladenen Objekten Markierungen hinzuzufügen, wählen Sie **Add tag (Tag hinzufügen)**. Geben Sie einen Tag-Namen in das Feld **Schlüssel** ein. Geben Sie einen Wert für das Tag ein.

Das Markieren von Objekten ermöglicht Ihnen, Speicher zu kategorisieren. Jeder Tag ist ein Schlüssel/Wert-Paar. Bei Schlüssel- und Tag-Werten wird die Groß-/Kleinschreibung berücksichtigt. Sie können über bis zu 10 Markierungen pro Objekt verfügen. Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen und die Tag-Werte bis zu 255 Unicode-Zeichen lang sein. Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

8. Um Metadaten hinzuzufügen, wählen Sie Add metadata (Metadaten hinzufügen).
 - a. Wählen Sie unter Typ die Option System defined (System definiert) oder User defined (Benutzerdefiniert) aus.

Für systemdefinierte Metadaten können Sie allgemeine HTTP Header wie Content-Type und Content-Disposition auswählen. Eine Liste systemdefinierter Metadaten und Informationen dazu, ob Sie Werte hinzufügen können, finden Sie unter [Systemdefinierte Objektmetadaten](#). Metadaten, die mit dem Präfix x-amz-met-a- beginnen, werden als benutzerdefinierte Metadaten behandelt. Benutzerdefinierte-Metadaten werden mit dem Objekt gespeichert und beim Download mit ihm zurückgegeben. Sowohl die Schlüssel als auch ihre Werte müssen den US-Standards entsprechen. ASCII Benutzerdefinierte Metadaten können bis zu 2 KB umfassen. Weitere Informationen zu systemdefinierten und benutzerdefinierten Metadaten finden Sie unter [Arbeiten mit Objekt-Metadaten](#).

- b. Wählen Sie für Key einen Schlüssel aus.
 - c. Geben Sie einen Wert für den Schlüssel ein.
9. Um Ihre Objekte hochzuladen, wählen Sie Upload (Hochladen) aus.

Amazon S3 lädt Ihr Objekt hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: status eine Erfolgsmeldung angezeigt.

10. Wählen Sie Exit (Beenden) aus.

Verwendung der AWS CLI

Sie können eine PUT-Anforderung zum Hochladen eines Objekts von bis zu 5 GB in einem einzigen Vorgang senden. Weitere Informationen finden Sie im [PutObject](#) Beispiel im AWS CLI Befehlsreferenz.

Mit dem REST API

Sie können REST Anfragen zum Hochladen eines Objekts senden. Sie können eine PUT-Anforderung senden, um Daten in einer einzigen Operation hochzuladen. Weitere Informationen finden Sie unter [PUTObjekt](#).

Verwendung der AWS SDKs

Sie können das AWS SDKsum Objekte in Amazon S3 hochzuladen. SDKsSie bieten Wrapper-Bibliotheken, mit denen Sie Daten einfach hochladen können. Weitere Informationen finden Sie in der [Liste der unterstützten SDKs Programme](#).

Hier sind einige Beispiele mit einigen ausgewähltenSDKs:

.NET

Das folgende C#-Codebeispiel erstellt zwei Objekte mit zwei PutObjectRequest-Anforderungen:

- Die erste PutObjectRequest-Anforderung speichert als Beispielobjektdaten eine Textzeichenfolge. Sie gibt zudem Bucket- und Objektschlüsselnamen an.
- Die zweite PutObjectRequest-Anforderung lädt eine Datei durch Angabe des Dateinamens hoch. Diese Anforderung gibt außerdem den ContentType-Header und optionale Objektmetadaten (einen Titel) ein.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDKfür .NET](#)in der AWS SDKfür .NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadObjectTest
    {
        private const string bucketName = "*** bucket name ***";
        // For simplicity the example creates two objects from the same file.
        // You specify key names for these objects.
```



```
private const string keyName1 = "**** key name for first object created ****";
private const string keyName2 = "**** key name for second object created
****";
private const string filePath = @"**** file path ****";
private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.EUWest1;

private static IAmazonS3 client;

public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    WritingAnObjectAsync().Wait();
}

static async Task WritingAnObjectAsync()
{
    try
    {
        // 1. Put object-specify only key name for the new object.
        var putRequest1 = new PutObjectRequest
        {
            BucketName = bucketName,
            Key = keyName1,
            ContentBody = "sample text"
        };

        PutObjectResponse response1 = await
client.PutObjectAsync(putRequest1);

        // 2. Put the object-set ContentType and add metadata.
        var putRequest2 = new PutObjectRequest
        {
            BucketName = bucketName,
            Key = keyName2,
            FilePath = filePath,
            ContentType = "text/plain"
        };

        putRequest2.Metadata.Add("x-amz-meta-title", "someTitle");
        PutObjectResponse response2 = await
client.PutObjectAsync(putRequest2);
    }
    catch (AmazonS3Exception e)
```

```
        {
            Console.WriteLine(
                "Error encountered ***. Message:'{0}' when writing an
object"
                , e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine(
                "Unknown encountered on server. Message:'{0}' when writing an
object"
                , e.Message);
        }
    }
}
```

Java

Das folgende Beispiel erstellt zwei Objekte. Das erste Objekt besitzt eine Textzeichenfolge als Daten, und das zweite Objekt ist eine Datei. Das Beispiel erstellt das erste Objekt durch Angabe des Bucket-Namen, des Objektschlüssels und von Textdaten direkt in einem Aufruf an `AmazonS3Client.putObject()`. Das Beispiel erstellt das zweite Objekt mittels einer `PutObjectRequest`, die den Bucket-Namen, den Objektschlüssel und den Dateipfad angibt. Die `PutObjectRequest` gibt zudem den `ContentType-Header` und Titelmetadaten an.

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte](#) im *AWS SDK for Java Entwicklerhandbuch*.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;
import java.io.IOException;

public class UploadObject {
```

```
public static void main(String[] args) throws IOException {
    Regions clientRegion = Regions.DEFAULT_REGION;
    String bucketName = "**** Bucket name ****";
    String stringObjKeyName = "**** String object key name ****";
    String fileObjKeyName = "**** File object key name ****";
    String fileName = "**** Path to file to upload ****";

    try {
        // This code expects that you have AWS credentials set up per:
        // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
        AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
            .withRegion(clientRegion)
            .build();

        // Upload a text string as a new object.
        s3Client.putObject(bucketName, stringObjKeyName, "Uploaded String
Object");

        // Upload a file as a new object with ContentType and title specified.
        PutObjectRequest request = new PutObjectRequest(bucketName,
fileObjKeyName, new File(fileName));
        ObjectMetadata metadata = new ObjectMetadata();
        metadata.setContentType("plain/text");
        metadata.addUserMetadata("title", "someTitle");
        request.setMetadata(metadata);
        s3Client.putObject(request);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

JavaScript

Im folgenden Beispiel wird eine vorhandene Datei in einen Amazon-S3-Bucket in einer spezifischen Region hochgeladen.

```
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new PutObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

PHP

Dieses Beispiel führt Sie durch die Verwendung von Klassen aus AWS SDK for PHP um ein Objekt mit einer Größe von bis zu 5 GB hochzuladen. Für größere Dateien müssen Sie den mehrteiligen API Upload-Vorgang verwenden. Weitere Informationen finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Example – Erstellen eines Objekts in einem Amazon-S3-Bucket, indem Daten hochgeladen werden

Im folgenden PHP Beispiel wird ein Objekt in einem angegebenen Bucket erstellt, indem Daten mit der `putObject()` Methode hochgeladen werden.

```
require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
```

```
$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key'     => $keyname,
        'Body'    => 'Hello, world!',
        'ACL'     => 'public-read'
    ]);

    // Print the URL to the object.
    echo $result['ObjectURL'] . PHP_EOL;
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

Das Tool AWS SDK for Ruby - Version 3 bietet zwei Möglichkeiten, ein Objekt auf Amazon S3 hochzuladen. Bei der ersten wird ein verwalteter Datei-Uploader verwendet, mit dem sich Dateien beliebiger Größe einfacher von einer Festplatte hochladen lassen. So verwenden Sie die Methode des verwalteten Datei-Uploaders:

1. Erstellen Sie eine Instance der `Aws::S3::Resource`-Klasse.
2. Verweisen Sie per Bucket-Name und Schlüssel auf das Ziel-Objekt. Objekte befinden sich in einem Bucket und haben eindeutige Schlüssel, die jedes Objekt identifizieren.
3. Rufen Sie `#upload_file` für das Objekt auf.

Example

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
    attr_reader :object
```

```
# @param object [Aws::S3::Object] An existing Amazon S3 object.
def initialize(object)
  @object = object
end

# Uploads a file to an Amazon S3 object by using a managed uploader.
#
# @param file_path [String] The path to the file to upload.
# @return [Boolean] True when the file is uploaded; otherwise false.
def upload_file(file_path)
  @object.upload_file(file_path)
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
#{e.message}"
  false
end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Der zweite Weg, den AWS SDK for Ruby - Version 3 kann ein Objekt mit der `#put` Methode von `hochladenAws::S3::Object`. Das ist praktisch, wenn das Objekt eine Zeichenfolge oder ein I/O-Objekt ist, bei dem es sich nicht um eine Datei auf einem Datenträger handelt. So verwenden Sie diese Methode:

1. Erstellen Sie eine Instance der `Aws::S3::Resource`-Klasse.
2. Verweisen Sie per Bucket-Name und Schlüssel auf das Ziel-Objekt.

3. Rufen Sie #put auf und übergeben Sie die Zeichenfolge oder das I/O-Objekt.

Example

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
    File.open(source_file_path, "rb") do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why:
#{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Verhindern Sie das Hochladen von Objekten mit identischen Schlüsselnamen

Sie können überprüfen, ob ein Objekt in Ihrem Bucket vorhanden ist, bevor Sie es mithilfe von bedingten Schreibvorgängen beim Hochladen erstellen. Dadurch kann verhindert werden, dass vorhandene Daten überschrieben werden. Durch bedingte Schreibvorgänge wird bestätigt, dass sich beim Hochladen kein Objekt mit demselben Schlüsselnamen bereits in Ihrem Bucket befindet.

Sie können bedingte Schreibvorgänge verwenden für [PutObject](#) oder [CompleteMultipartUpload](#)Anfragen.

Weitere Informationen zu bedingten Anfragen finden Sie unter [Fügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu](#).

Hochladen und Kopieren von Objekten mit mehrteiligen Uploads

Mit dem mehrteiligen Upload können Sie ein einzelnes Objekt als Satz aus mehreren Teilen hochladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt. Wenn Ihre Objektgröße 100 MB erreicht, sollten Sie in der Regel mehrteilige Uploads verwenden, anstatt das Objekt in einem einzigen Vorgang hochzuladen.

Die Nutzung mehrteiliger Uploads bietet die folgenden Vorteile:

- Verbesserter Durchsatz – Sie können die Teile parallel hochladen, um den Durchsatz zu erhöhen.
- Schnelle Wiederherstellung bei Netzwerkproblemen – Die kleinere Teilegröße minimiert die Auswirkungen eines Neustarts eines fehlgeschlagenen Uploads aufgrund eines Netzwerkfehlers.
- Anhalten und Fortsetzen von Objekt-Uploads – Sie können Objektteile mit der Zeit hochladen. Nachdem Sie einen mehrteiligen Upload initiiert haben, gibt es kein Ablaufdatum. Sie müssen den mehrteiligen Upload ausdrücklich abschließen oder abbrechen.
- Starten Sie einen Upload, bevor Sie die endgültige Objektgröße kennen – Sie können ein Objekt hochladen, während Sie es noch erstellen.

Sie sollten den mehrteiligen Upload wie folgt verwenden:

- Wenn Sie große Objekte über ein stabiles Netzwerk mit hoher Bandbreite hochladen, können Sie einen mehrteiligen Upload verwenden, um die Nutzung der verfügbaren Bandbreite zu maximieren. Hierzu laden Sie Objektteile parallel hoch, um von einer Multi-Threading-Leistung zu profitieren.
- Wenn Sie einen Upload über ein instabiles Netzwerk ausführen, können Sie einen mehrteiligen Upload verwenden, um die Resilienz in Bezug auf Netzwerkfehler durch Vermeidung von Neustarts der Uploads zu vermeiden. Wenn Sie mehrteilige Uploads verwenden, müssen Sie nur die Teile erneut hochladen, deren Upload unterbrochen wurde. Sie müssen nicht das gesamte Objekt von Anfang an neu hochladen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#). Weitere Informationen zur Verwendung mehrteiliger Uploads mit S3 Express One Zone und Verzeichnis-Buckets finden Sie unter [Mehrteilige Uploads mit Verzeichnis-Buckets verwenden](#).

Mehrteiliger Upload-Prozess

Der mehrteilige Upload ist ein Prozess in drei Schritten: Sie beginnen den Upload, laden die Teile des Objekts hoch und schließen den mehrteiligen Upload ab, wenn alle Teile hochgeladen wurden. Nach dem vollständigen Eingang der Daten aus dem mehrteiligen Upload erstellt Amazon S3 das Objekt anhand der hochgeladenen Teile. Dann können Sie auf das Objekt wie auf jedes andere Objekt in Ihrem Bucket zugreifen.

Sie können alle laufenden mehrteiligen Uploads auflisten oder eine Liste der Teile anfordern, die Sie für einen bestimmten Multipart-Upload hochgeladen haben. Alle Vorgänge werden in diesem Abschnitt erklärt.

Initiieren des mehrteiligen Uploads

Wenn Sie eine Anfrage zum Initiieren eines mehrteiligen Uploads senden, gibt Amazon S3 eine Antwort mit einer Upload-ID zurück, als eindeutige Kennung für den Multipart-Upload. Sie müssen diese Upload-ID immer angeben, wenn Sie Teile hochladen, die Teile auflisten, einen Upload abschließen oder einen Upload abbrechen. Wenn Sie Metadaten bereitstellen möchten, die das hochzuladende Objekt beschreiben, müssen sie in der Anforderung auf Initiierung des mehrteiligen Uploads angegeben werden.

Teile hochladen

Beim Hochladen eines Teils müssen Sie zusätzlich zur Upload-ID eine Teilenummer angeben. Sie können jede Teilenummer zwischen 1 und 10.000 wählen. Die Teilenummer identifiziert eindeutig einen Teil und seine Position im Objekt, das Sie hochladen. Die von Ihnen gewählte Teilenummer muss nicht fortlaufend sein (möglich sind z. B. 1, 5 und 14). Wenn Sie einen neuen Teil mit derselben Teilenummer hochladen wie bereits einmal zuvor, wird der früher hochgeladene Teil überschrieben.

Wenn Sie ein Teil hochladen, gibt Amazon S3 ein Entitäts-Tag (ETag) für das Teil als Header in der Antwort zurück. Für jeden Upload eines Teils müssen Sie die Artikelnummer und den ETag Wert aufzeichnen. Sie müssen diese Werte in die spätere Anforderung einschließen, um den mehrteiligen Upload abzuschließen. Jedes Teil wird zum ETag Zeitpunkt des Uploads sein eigenes haben. Sobald der mehrteilige Upload jedoch abgeschlossen ist und alle Teile konsolidiert sind, werden alle Teile ETag als Prüfsumme von Prüfsummen unter eins angezeigt.

Note

Nachdem Sie einen mehrteiligen Upload gestartet und einen oder mehrere Teile hochgeladen haben, müssen Sie den mehrteiligen Upload abschließen oder abbrechen, damit keine Gebühren für die Speicherung der hochgeladenen Teile anfallen. Erst nach dem Abschluss oder Abbruch eines mehrteiligen Uploads gibt Amazon S3 den Speicher für die Teile frei und stoppt die Berechnung von Gebühren für die Speicherung der Teile.

Wenn Sie einen mehrteiligen Upload abbrechen, können Sie mit dieser Upload-ID keine Teile mehr hochladen. Wenn der mehrteilige Upload abgebrochen wird, während Teile hochgeladen werden, können diese Uploads auch nach dem Abbruch des Uploads erfolgreich abgeschlossen werden oder fehlschlagen. Um sicherzustellen, dass der von allen Teilen verbrauchte Speicherplatz freigegeben wird, dürfen Sie einen mehrteiligen Upload erst dann abbrechen, wenn alle Uploads abgeschlossen wurden.

Abschließen eines mehrteiligen Uploads

Wenn Sie einen mehrteiligen Upload abschließen, erstellt Amazon S3 ein Objekt, indem die Teile in aufsteigender Reihenfolge auf Grundlage der Teilenummer verkettet werden. Wenn Sie Metadaten für das Objekt bei der Initiierung des mehrteiligen Uploads bereitgestellt haben, verknüpft Amazon S3 die Metadaten mit dem Objekt. Nach einer erfolgreich ausgeführten Abschlussanforderung sind die Teile nicht mehr vorhanden.

Ihre vollständige mehrteilige Upload-Anfrage muss die Upload-ID und eine Liste der beiden Artikelnummern und der entsprechenden Werte enthalten. ETag Die Amazon S3 S3-Antwort enthält eineETag, die die kombinierten Objektdaten eindeutig identifiziert. Dies ETag ist nicht unbedingt ein MD5 Hash der Objektdaten.

Beispielaufrufe mehrteiliger Uploads

Für dieses Beispiel nehmen wir an, dass Sie einen mehrteiligen Upload für eine 100-GB-Datei generieren. In diesem Fall hätten Sie die folgenden API Aufrufe für den gesamten Prozess. Es würde insgesamt 1002 API Anrufe geben.

- Ein [CreateMultipartUpload](#)-Aufruf zum Starten des Prozesses
- 1 000 individuelle [UploadPart](#)-Aufrufe, mit denen jeweils ein Teil von 100 MB für die Gesamtgröße von 100 GB hochgeladen wird
- Ein [CompleteMultipartUpload](#)-Aufruf zum Beenden des Prozesses

Auflistungen mehrteiliger Uploads

Sie können alle Teile eines bestimmten Multipart-Uploads oder alle laufenden mehrteiligen Uploads auflisten. Die Operation für die Teileauflistung gibt die Teileinformationen zurück, die Sie für einen bestimmten mehrteiligen Upload hochgeladen haben. Für jeden Abruf einer Teileauflistung gibt Amazon S3 die Teileinformationen für einen angegebenen mehrteiligen Upload bis zu maximal 1.000 Teilen zurück. Wenn im Multipart-Upload mehr als 1.000 Teile vorhanden sind, müssen Sie eine Reihe von Anforderungen auf Teileauflistung senden, um alle Teile abzurufen. Beachten Sie, dass die zurückgegebene Teileauflistung keine Teile enthält, die noch nicht vollständig hochgeladen wurden. Bei Verwendung der Operation Mehrteilige Uploads auflisten können Sie eine Liste aller mehrteiligen Uploads erhalten, die sich in Bearbeitung befinden.

Ein mehrteiliger Upload in Verarbeitung ist ein Upload, den Sie gestartet haben, der aber noch nicht abgeschlossen ist oder abgebrochen wurde. Jeder Anforderung gibt bis zu 1.000 mehrteilige Uploads zurück. Wenn mehr als 1 000 mehrteilige Uploads vorhanden sind, müssen Sie zusätzliche Anforderungen senden, um die verbleibenden mehrteiligen Uploads abzurufen. Verwenden Sie die zurückgegebene Liste nur zur Überprüfung. Verwenden Sie das Ergebnis dieser Auflistung nicht, wenn Sie eine Anforderung für den Abschluss eines mehrteiligen Uploads senden. Pflegen Sie stattdessen Ihre eigene Liste mit den Artikelnummern, die Sie beim Hochladen von Teilen angegeben haben, und den entsprechenden ETag Werten, die Amazon S3 zurückgibt.

Prüfsummen mit mehrteiligen Upload-Operationen

Wenn Sie ein Objekt auf Amazon S3 hochladen, können Sie einen Prüfsummenalgorithmus angeben, den Amazon S3 verwenden soll. Amazon S3 verwendet MD5 standardmäßig zur Überprüfung der Datenintegrität. Sie können jedoch einen zusätzlichen Prüfsummenalgorithmus angeben, der verwendet werden soll. Bei Verwendung MD5 berechnet Amazon S3 die Prüfsumme des gesamten mehrteiligen Objekts, nachdem der Upload abgeschlossen ist. Diese Prüfsumme ist keine Prüfsumme des gesamten Objekts, sondern eine Prüfsumme der Prüfsummen für jeden einzelnen Teil.

Wenn Sie Amazon S3 anweisen, zusätzliche Prüfsummen zu verwenden, berechnet Amazon S3 den Prüfsummenwert für jeden Teil und speichert die Werte. Sie können API oder verwenden, SDK um den Prüfsummenwert für einzelne Teile abzurufen, indem Sie oder verwenden. `GetObject` `HeadObject` Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie `ListParts` verwenden.

Important

Wenn Sie einen mehrteiligen Upload mit zusätzlichen Prüfsummen verwenden, müssen die mehrteiligen Teilenummern aufeinander folgende Teilenummern sein. Wenn Sie zusätzliche Prüfsummen verwenden und versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen Fehler. `HTTP 500 Internal Server Error`

Weitere Informationen zur Funktionsweise von Prüfsummen mit mehrteiligen Objekten finden Sie unter [Überprüfung der Objektintegrität](#).

Gleichzeitige mehrteilige Upload-Vorgänge

In einer verteilten Entwicklungsumgebung ist es für Ihre Anwendung möglich, mehrere Updates gleichzeitig für dasselbe Objekt zu initiieren. Ihre Anwendung kann möglicherweise mehrere Multipart-Uploads mit demselben Objektschlüssel initiieren. Für jeden dieser Uploads kann Ihre Anwendung Teile hochladen und eine Anfrage auf Abschluss des Uploads an Amazon S3 senden, um das Objekt zu erstellen. Wenn die Buckets die S3-Versioning aktiviert haben, wird beim Abschluss eines mehrteiligen Uploads immer eine neue Version erstellt. Wenn Sie mehrere mehrteilige Uploads initiieren, die denselben Objektschlüssel in einem Bucket mit aktivierter Versionierung verwenden, wird die aktuelle Version des Objekts anhand der zuletzt gestarteten Version des Objekts bestimmt (). `createdDate` Nehmen wir zum Beispiel an, dass Sie um 10:00

Uhr eine CreateMultipartUpload Anfrage für ein Objekt starten. Dann reichen Sie um 11:00 Uhr eine zweite CreateMultipartUpload Anfrage für dasselbe Objekt ein. Da die zweite Anfrage zuletzt eingereicht wurde, entspricht das mit der Anfrage um 11:00 Uhr hochgeladene Objekt der aktuellen Version, auch wenn der erste Upload nach dem zweiten abgeschlossen ist. Bei Buckets, für die keine Versioning aktiviert ist, kann es sein, dass zwischen den Zeitpunkten der Initiierung bis zum Abschluss eines Multipart-Uploads eine andere Anforderung vorrangig ist.

Note

Es ist möglich, dass eine andere Anforderung zwischen dem Zeitpunkt der Initiierung und jenem des Abschlusses eines Multipart-Uploads stattgefunden hat. Wenn z. B. ein anderer Vorgang einen Schlüssel löscht, nachdem Sie einen mehrteiligen Upload mit diesem Schlüssel initiiert haben, aber bevor Sie ihn abschließen, kann die Antwort für den Abschluss des Multipart-Uploads möglicherweise eine erfolgreiche Objekterstellung anzeigen, ohne dass Sie das Objekt je zu Gesicht bekommen haben.

Verhindern Sie, dass beim mehrteiligen Upload Objekte mit identischen Schlüsselnamen hochgeladen werden

Sie können überprüfen, ob ein Objekt in Ihrem Bucket vorhanden ist, bevor Sie es mithilfe von bedingten Schreibvorgängen beim Hochladen erstellen. Dadurch kann verhindert werden, dass vorhandene Daten überschrieben werden. Durch bedingte Schreibvorgänge wird bestätigt, dass sich beim Hochladen kein Objekt mit demselben Schlüsselnamen bereits in Ihrem Bucket befindet.

Sie können bedingte Schreibvorgänge verwenden für [PutObject](#) oder [CompleteMultipartUpload](#) Anfragen.

Weitere Informationen zu bedingten Anfragen finden Sie unter [Fügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu](#).

Mehrteiliger Upload und Preise

Nachdem Sie einen mehrteiligen Upload gestartet haben, behält Amazon S3 alle Teile bei, bis Sie den Upload abschließen oder abbrechen. Während seiner gesamten Lebensdauer werden Ihnen der gesamte Speicher, die Bandbreite und die Anforderungen für diesen mehrteiligen Upload und die zugehörigen Teile in Rechnung gestellt.

Diese Teile werden gemäß der Speicherklasse berechnet, die beim Hochladen der Teile angegeben wurde. Eine Ausnahme bilden Teile, die in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive hochgeladen wurden. In Bearbeitung befindliche mehrteilige Teile für die Speicherklasse S3 Glacier Flexible Retrieval werden PUT bis zum Abschluss des Uploads als S3 Glacier Flexible Retrieval Staging Storage zu den Speichertarifen S3 Standard abgerechnet. Darüber hinaus werden sowohl als auch zu den CreateMultipartUpload S3-Standardtarifen UploadPart abgerechnet. Nur die CompleteMultipartUpload Anfrage wird zum Tarif S3 Glacier Flexible Retrieval abgerechnet. In ähnlicher Weise werden in Bearbeitung befindliche mehrteilige Teile für eine Speicherklasse PUT bis zur S3 Glacier Deep Archive Archive-Speicherklasse als S3 Glacier Flexible Retrieval Staging Storage zu den Speichertarifen S3 Standard abgerechnet, bis der Upload abgeschlossen ist, wobei nur die CompleteMultipartUpload Anfrage zu den S3 Glacier Deep Archive Archive-Tarifen berechnet wird.

Wenn Sie den mehrteiligen Upload abbrechen, löscht Amazon S3 Upload-Artefakte und alle Teile, die Sie hochgeladen haben. Die Berechnung wird eingestellt. Für das Löschen unvollständiger mehrteiliger Uploads fallen keine Gebühren für vorzeitiges Löschen an, unabhängig von der angegebenen Speicherklasse. Weitere Informationen zu Preisen finden Sie unter [Amazon-S3-Preise](#).

Note

Wir empfehlen, eine Lebenszyklusregel zu konfigurieren, durch die unvollständige mehrteilige Uploads nach einer bestimmten Anzahl von Tagen mit der Aktion `AbortIncompleteMultipartUpload` gelöscht werden, um Ihre Speicherkosten gering zu halten. Weitere Informationen zum Erstellen einer Lebenszyklusregel zum Löschen unvollständiger mehrteiliger Uploads finden Sie unter [Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads](#).

API-Unterstützung für mehrteilige Uploads

Diese Bibliotheken bieten eine hohe Abstraktion, wodurch der mehrteilige Upload von Objekten sehr einfach wird. Wenn Ihre Anwendung dies jedoch erfordert, können Sie das REST API direkt verwenden. In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird das REST API für mehrteilige Uploads beschrieben.

Für eine exemplarische Vorgehensweise zum mehrteiligen Upload, die verwendet AWS Lambda-Funktionen, siehe [Hochladen großer Objekte auf Amazon S3 mithilfe der mehrteiligen Upload- und Übertragungsbeschleunigung](#).

- [Multipart-Upload erstellen](#)
- [Upload Part](#)
- [Teil hochladen \(kopieren\)](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Abort Multipart Upload](#)
- [Teile auflisten](#)
- [List Multipart Uploads](#)

AWS Command Line Interface Unterstützung für mehrteiligen Upload

Die folgenden Themen finden Sie im AWS Command Line Interface beschreibt die Operationen für den mehrteiligen Upload.

- [Initiieren eines mehrteiligen Uploads](#)
- [Upload Part](#)
- [Teil hochladen \(kopieren\)](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Abort Multipart Upload](#)
- [Teile auflisten](#)
- [List Multipart Uploads](#)

AWS SDK-Unterstützung für mehrteilige Uploads

Sie können eine verwenden AWS SDKsum ein Objekt in Teilen hochzuladen. Für eine Liste AWS SDKsunterstützt durch API Aktion siehe:

- [Multipart-Upload erstellen](#)
- [Upload Part](#)
- [Teil hochladen \(kopieren\)](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Abort Multipart Upload](#)

- [Teile auflisten](#)
- [List Multipart Uploads](#)

Mehrteiliger Upload API und Berechtigungen

Sie müssen über die erforderlichen Berechtigungen verfügen, um die Multipart-Upload-Vorgänge zu verwenden. Sie können Zugriffskontrolllisten (ACLs), die Bucket-Richtlinie oder die Benutzerrichtlinie verwenden, um Einzelpersonen Berechtigungen zur Ausführung dieser Operationen zu gewähren. In der folgenden Tabelle sind die erforderlichen Berechtigungen für verschiedene mehrteilige Upload-Operationen aufgeführt. ACLs, wenn Sie eine Bucket-Richtlinie oder eine Benutzerrichtlinie verwenden.

Aktion	Erforderliche Berechtigungen
Multipart-Upload erstellen	<p>Sie müssen die Aktion <code>s3:PutObject</code> für ein Objekt ausführen können, um einen Multipart-Upload zu erstellen.</p> <p>Der Bucket-Eigentümer kann anderen Prinzipalen erlauben, die Aktion <code>s3:PutObject</code> auszuführen.</p>
Initiieren eines mehrteiligen Uploads	<p>Sie müssen die Aktion <code>s3:PutObject</code> für ein Objekt ausführen können, um einen Multipart-Upload zu initiieren.</p> <p>Der Bucket-Eigentümer kann anderen Prinzipalen erlauben, die Aktion <code>s3:PutObject</code> auszuführen.</p>
Initiator	<p>Containerelement, das identifiziert, wer den Multipart-Upload initiiert hat. Wenn der Initiator ein ist AWS-Konto, stellt dieses Element dieselben Informationen bereit wie das Owner-Element. Wenn der Initiator ein IAM Benutzer ist, stellt dieses Element den Benutzer ARN und den Anzeigenamen bereit.</p>
Upload Part	<p>Sie müssen die Aktion <code>s3:PutObject</code> für ein Objekt ausführen können, um einen Teil hochzuladen.</p> <p>Der Bucket-Eigentümer muss dem Initiator erlauben, die Aktion <code>s3:PutObject</code> für ein Objekt auszuführen, damit der Initiator einen Teil für dieses Objekt hochladen kann.</p>

Aktion	Erforderliche Berechtigungen
Teil hochladen (kopieren)	<p>Sie müssen die Aktion <code>s3:PutObject</code> für ein Objekt ausführen können, um einen Teil hochzuladen. Da Sie einen Teil eines vorhandenen Objekts hochladen, müssen Sie die Berechtigung <code>s3:GetObject</code> für das Quellobjekt besitzen.</p> <p>Damit der Initiator einen Teil eines Objekts hochladen kann, muss der Bucket-Besitzer dem Initiator eine Genehmigung für die Ausführung der Aktion <code>s3:PutObject</code> für das Objekt erteilen.</p>
Abschließen eines mehrteiligen Uploads	<p>Sie müssen die Aktion <code>s3:PutObject</code> für ein Objekt ausführen können, um einen Multipart-Upload abzuschließen.</p> <p>Der Bucket-Eigentümer muss dem Initiator erlauben, die Aktion <code>s3:PutObject</code> für ein Objekt auszuführen, damit der Initiator den mehrteiligen Upload für dieses Objekt abschließen kann.</p>
Mehrteiligen Upload abbrechen	<p>Sie müssen die Aktion <code>s3:AbortMultipartUpload</code> ausführen dürfen, um einen mehrteiligen Upload abzurechnen.</p> <p>Standardmäßig dürfen der Bucket-Besitzer und der Initiator des mehrteiligen Uploads diese Aktion im Rahmen der IAM Bucket-Richtlinien ausführen. Wenn der Initiator ein IAM Benutzer ist, ist dieser Benutzer AWS-Konto darf diesen mehrteiligen Upload auch beenden. Bei VPC Endpunktrichtlinien erhält der Initiator des mehrteiligen Uploads nicht automatisch die Erlaubnis, die Aktion auszuführen. <code>s3:AbortMultipartUpload</code></p> <p>Zusätzlich zu diesen Standardberechtigungen, kann der Bucket-Eigentümer anderen Prinzipalen erlauben, die Aktion <code>s3:AbortMultipartUpload</code> auszuführen. Der Bucket-Eigentümer kann jedem Prinzipal verbieten, die Aktion <code>s3:AbortMultipartUpload</code> auszuführen.</p>

Aktion	Erforderliche Berechtigungen
Teile auflisten	<p>Sie müssen die Aktion <code>s3:ListMultipartUploadParts</code> ausführen können, um Teileaflistungen in einem mehrteiligen Upload anzufragen.</p> <p>Standardmäßig kann der Bucket-Eigentümer Teileaflistungen von Multipart-Uploads in seinem Bucket anfordern. Der Initiator des mehrteiligen Uploads hat die Berechtigung, Teileaflistungen des spezifischen Multipart-Uploads aufzulisten. Wenn der Initiator des mehrteiligen Uploads ein Benutzer ist, IAM AWS-Konto IAMDer Benutzer, der diesen Benutzer kontrolliert, ist auch berechtigt, Teile dieses Uploads aufzulisten.</p> <p>Zusätzlich zu diesen Standardberechtigungen, kann der Bucket-Eigentümer anderen Prinzipalen erlauben, die Aktion <code>s3:ListMultipartUploadParts</code> auszuführen. Der Bucket-Eigentümer kann jedem Prinzipal auch verbieten, die Aktion <code>s3:ListMultipartUploadParts</code> auszuführen.</p>
List Multipart Uploads	<p>Sie müssen die Aktion <code>s3:ListBucketMultipartUploads</code> für einen Bucket ausführen können, um eine Multipart-Uploads-Auflistung für diesen Bucket anzufragen.</p> <p>Zusätzlich zu diesen Standardberechtigungen, kann der Bucket-Eigentümer anderen Prinzipalen erlauben, die Aktion <code>s3:ListBucketMultipartUploads</code> für ein Bucket auszuführen.</p>

Aktion	Erforderliche Berechtigungen
<p>AWS KMS</p> <p>Die entsprechenden Berechtigungen verschlüsseln und entschlüsseln</p>	<p>Um einen mehrteiligen Upload mit Verschlüsselung mit einem durchzuführen AWS Key Management Service (AWS KMS) KMS -Schlüssel, der Anforderer muss über die erforderlichen Berechtigungen für den Schlüssel <code>kms:Decrypt</code> und die damit <code>kms:GenerateDataKey</code> verbundenen Aktionen verfügen. Der Anforderer muss außerdem über Berechtigungen für die <code>kms:GenerateDataKey</code> Aktion für den verfügen CreateMultipartUpload API. Dann benötigt der Anforderer Berechtigungen für die <code>kms:Decrypt</code> Aktion auf der UploadPart und UploadPartCopy APIs. Diese Berechtigungen sind erforderlich, da Amazon S3 Daten aus den verschlüsselten Teilen der Datei entschlüsseln und lesen muss, bevor es den Multipart-Upload vornehmen kann.</p> <p>Wenn sich Ihr IAM Benutzer oder Ihre Rolle in derselben befindet AWS-Konto wie der KMS Schlüssel, dann müssen Sie über diese Berechtigungen für die Schlüsselrichtlinie verfügen. Wenn Ihr IAM Benutzer oder Ihre Rolle zu einem anderen Konto als dem KMS Schlüssel gehört, müssen Sie über die erforderlichen Berechtigungen sowohl für die Schlüsselrichtlinie als auch für Ihren IAM Benutzer oder Ihre Rolle verfügen.</p>

Informationen zur Beziehung zwischen ACL Berechtigungen und Berechtigungen in Zugriffsrichtlinien finden Sie unter [Zuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen](#).

Informationen zu IAM Benutzern, Rollen und bewährten Methoden finden Sie im Benutzerhandbuch unter [IAM Identitäten \(Benutzer, Benutzergruppen und Rollen\)](#). IAM

Themen

- [Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads](#)
- [Hochladen eines Objekts mit Multipart-Upload](#)
- [Laden eines Verzeichnisses mit der High-Level-Version hoch. NET TransferUtility Klasse](#)
- [Auflisten von mehrteiligen Uploads](#)
- [Verfolgen eines mehrteiligen Uploads](#)
- [Abbrechen eines mehrteiligen Uploads](#)
- [Kopieren eines Objekts mit Multipart-Upload](#)
- [Einschränkungen mehrteiliger Amazon-S3-Uploads](#)

Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads

Als bewährte Methode empfehlen wir Ihnen, eine Lebenszyklusregel mit der Aktion `AbortIncompleteMultipartUpload` zu konfigurieren, um Ihre Speicherkosten zu minimieren. Weitere Informationen zum Abbruch eines mehrteiligen Uploads finden Sie unter [Abbrechen eines mehrteiligen Uploads](#).

Amazon S3 unterstützt eine Bucket-Lebenszyklusregel, mit der Sie Amazon S3 anweisen können, unvollständige mehrteilige Uploads abzubrechen, die nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wenn ein mehrteiliger Upload nicht innerhalb des angegebenen Zeitrahmens abgeschlossen wird, wird er für eine Abbruchoperation zugelassen. Amazon S3 bricht den mehrteiligen Upload dann ab und löscht alle diesem mehrteiligen Upload zugeordneten Teile. Diese Regel gilt sowohl für bestehende mehrteilige Uploads als auch für solche, die Sie später erstellen.

Das folgende Beispiel zeigt eine Lebenszykluskonfiguration, die eine Regel mit der Aktion `AbortIncompleteMultipartUpload` spezifiziert.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>7</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

In dem Beispiel gibt die Regel keinen Wert für das `Prefix`-Element ([Objektschlüsselnamen-Präfix](#)) an. Daher gilt die Regel für alle Objekte im Bucket, für die Sie mehrteilige Uploads initiiert haben. Alle mehrteiligen Uploads, die initiiert wurden und nicht innerhalb von sieben Tagen abgeschlossen wurden, werden für eine Abbruchoperation zugelassen. Die Abbruchaktion hat keine Auswirkung auf abgeschlossene mehrteilige Uploads.

Weitere Informationen zur Bucket-Lebenszykluskonfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Note

Wenn der mehrteilige Upload innerhalb der in der Regel angegebenen Anzahl von Tagen abgeschlossen wurde, gilt die Lebenszyklusaktion `AbortIncompleteMultipartUpload` nicht (das heißt, Amazon S3 führt keine Aktion aus). Außerdem gilt diese Aktion nicht für Objekte. Durch diese Lebenszyklus-Aktion werden keine Objekte gelöscht. Darüber hinaus fallen keine Gebühren für das vorzeitige Löschen für den S3-Lebenszyklus an, wenn Sie Teile von unvollständigen mehrteiligen Uploads entfernen.

Verwenden der S3-Konsole

Um unvollständige mehrteilige Uploads automatisch zu verwalten, können Sie die S3-Konsole verwenden, um eine Lebenszyklusrichtlinie zu erstellen und unvollständige mehrteilige Upload-Bytes aus dem Bucket nach einer bestimmten Anzahl von Tagen ablaufen zu lassen. Im folgenden Verfahren wird gezeigt, wie Sie eine Lebenszyklusregel hinzufügen, um unvollständige mehrteilige Uploads nach 7 Tagen zu löschen. Weitere Informationen über das Hinzufügen von Lebenszyklusregeln finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#).

So fügen Sie eine Lebenszyklusregel hinzu, um unvollständige mehrteilige Uploads, die älter als 7 Tage sind, abzurechnen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Lebenszyklusregel erstellen möchten.
3. Wählen Sie den Tab Management (Verwaltung) und dann die Option Create lifecycle rule (Lebenszyklusregel erstellen).
4. Geben Sie unter Lifecycle rule name (Name der Lebenszyklusregel) einen Namen für Ihre Regel ein.

Der Name muss innerhalb des Buckets eindeutig sein.

5. Wählen Sie den Umfang der Lebenszyklusregel:
 - Wenn Sie eine Lebenszyklusregel für alle Objekte mit einem bestimmten Präfix erstellen möchten, wählen Sie Limit the scope of this rule using one or more filters (Den Geltungsbereich dieser Regel mithilfe eines oder mehrerer Filter einschränken) aus und geben Sie das Präfix im Feld Prefix (Präfix) ein.

- Wenn Sie diese Lebenszyklusregel auf alle Objekte im Bucket anwenden möchten, wählen Sie `This rule applies to all objects in the bucket` (Diese Regel gilt für alle Objekte in dem Bucket) aus und klicken Sie auf `I acknowledge that this rule applies to all objects in the bucket` (Ich bestätige, dass diese Regel für alle Objekte in dem Bucket gilt).
6. Wählen Sie unter `Lifecycle rule actions` (Lebenszyklusregelaktionen) die Option `Delete expired object delete markers or incomplete multipart uploads` (Abgelaufene Objektlöschmarken oder unvollständige mehrteilige Uploads löschen) aus.
 7. Wählen Sie unter `Delete expired object delete markers or incomplete multipart uploads` (Abgelaufene Objektlöschmarken oder unvollständige mehrteilige Uploads löschen) die Option `Delete incomplete multipart uploads` (Unvollständige mehrteilige Uploads löschen) aus.
 8. Geben Sie im Feld `Number of days` (Anzahl der Tage) die Anzahl der Tage ein, nach denen unvollständige mehrteilige Uploads gelöscht werden sollen (in diesem Beispiel 7 Tage).
 9. Wählen Sie `Regel erstellen` aus.

Verwendung der AWS CLI

Folgendes `put-bucket-lifecycle-configuration` AWS Command Line Interface (AWS CLI) Befehl) fügt die Lebenszykluskonfiguration für den angegebenen Bucket hinzu. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api put-bucket-lifecycle-configuration \
  --bucket amzn-s3-demo-bucket1 \
  --lifecycle-configuration filename-containing-lifecycle-configuration
```

Das folgende Beispiel zeigt, wie Sie eine Lebenszyklusregel hinzufügen, um unvollständige mehrteilige Uploads abzubrechen, indem Sie AWS CLI. Es enthält ein Beispiel für eine JSON Lebenszykluskonfiguration zum Abbrechen unvollständiger mehrteiliger Uploads, die älter als 7 Tage sind.

Um die CLI Befehle in diesem Beispiel zu verwenden, ersetzen Sie die durch Ihre Informationen *user input placeholders*.

So fügen Sie eine Lebenszyklusregel hinzu, um unvollständige mehrteilige Uploads abzubrechen

1. Richten Sie das ein AWS CLI Entsprechende Anweisungen finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

- Speichern Sie die folgende Beispiel-Lebenszykluskonfiguration in einer Datei (zum Beispiel *lifecycle.json*). Diese Beispielkonfiguration spezifiziert ein leeres Präfix und gilt daher für alle Objekte im Bucket. Sie können ein Präfix angeben, um die Konfiguration auf eine Teilmenge von Objekten zu beschränken.

```
{
  "Rules": [
    {
      "ID": "Test Rule",
      "Status": "Enabled",
      "Filter": {
        "Prefix": ""
      },
      "AbortIncompleteMultipartUpload": {
        "DaysAfterInitiation": 7
      }
    }
  ]
}
```

- Führen Sie den folgenden CLI Befehl aus, um diese Lebenszykluskonfiguration für Ihren Bucket festzulegen.

```
aws s3api put-bucket-lifecycle-configuration \
--bucket amzn-s3-demo-bucket1 \
--lifecycle-configuration file://lifecycle.json
```

- Um zu überprüfen, ob die Lebenszykluskonfiguration für Ihren Bucket festgelegt wurde, rufen Sie die Lebenszykluskonfiguration mit dem folgenden `get-bucket-lifecycle`-Befehl ab.

```
aws s3api get-bucket-lifecycle \
--bucket amzn-s3-demo-bucket1
```

- Verwenden Sie zum Löschen der Lebenszykluskonfiguration den folgenden `delete-bucket-lifecycle`-Befehl.

```
aws s3api delete-bucket-lifecycle \
--bucket amzn-s3-demo-bucket1
```

Hochladen eines Objekts mit Multipart-Upload

Sie können den Multipart-Upload verwenden, um ein einzelnes Objekt programmgesteuert auf Amazon S3 hochzuladen.

Weitere Informationen finden Sie in den folgenden Abschnitten.

Verwenden der S3-Konsole

Sie können beliebige Dateitypen – Bilder, Backups, Daten, Filme usw. – in einen S3-Bucket hochladen. Die maximale Größe einer Datei, die Sie über die Amazon-S3-Konsole hochladen können, ist 160 GB. Um eine Datei mit mehr als 160 GB hochzuladen, verwenden Sie AWS Command Line Interface (AWS CLI), AWS SDKs oder Amazon S3 REST API.

Anweisungen zum Hochladen eines Objekts über AWS Management Console, finden Sie unter [Objekte hochladen](#).

Verwendung der AWS CLI

Die folgenden Abschnitte in AWS Command Line Interface (AWS CLI) beschreiben die Operationen für den mehrteiligen Upload.

- [Initiieren eines mehrteiligen Uploads](#)
- [Upload Part](#)
- [Teil hochladen \(kopieren\)](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Abort Multipart Upload](#)
- [Teile auflisten](#)
- [List Multipart Uploads](#)

Sie können den auch verwenden REST API, um Ihre eigenen REST Anfragen zu stellen, oder Sie können eine der AWS SDKs. Weitere Informationen zu den finden REST API Sie unter [Mit dem REST API](#). Weitere Informationen über die finden SDKs Sie unter [Hochladen eines Objekts mit Multipart-Upload](#).

Mit dem REST API

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird das REST API für mehrteilige Uploads beschrieben.

- [Initiieren eines mehrteiligen Uploads](#)
- [Upload Part](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Mehrteiligen Upload abbrechen](#)
- [Teile auflisten](#)
- [List Multipart Uploads](#)

Verwendung der AWS SDKs

Beispiele dafür, wie Sie einen mehrteiligen Upload durchführen können, finden Sie unter AWS SDKs, siehe. [Führen Sie einen mehrteiligen Upload eines Amazon S3 S3-Objekts mit einem AWS SDK](#)

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Verwendung der AWS SDKs(auf hohem NiveauAPI)

Etwas AWS SDKs stellt eine allgemeine Version zur API Verfügung, die den mehrteiligen Upload vereinfacht, indem die verschiedenen zum Abschließen eines mehrteiligen Uploads erforderlichen API Operationen zu einem einzigen Vorgang zusammengefasst werden. Weitere Informationen finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Wenn Sie mehrteilige Uploads anhalten und fortsetzen müssen, die Teilgrößen während des Uploads variieren müssen oder die Größe der Daten nicht im Voraus kennen, verwenden Sie die Low-Level-Methoden. API Die API Low-Level-Methoden für mehrteilige Uploads bieten zusätzliche Funktionen. Weitere Informationen finden Sie unter. [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#)

Java

Beispiele dafür, wie Sie einen mehrteiligen Upload mit Java durchführen, finden Sie unter. SDK [Führen Sie einen mehrteiligen Upload eines Amazon S3 S3-Objekts mit einem AWS SDK](#)


Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

.NET

Um eine Datei in einen S3-Bucket hochzuladen, verwenden Sie die Klasse `TransferUtility`. Beim Hochladen von Daten aus einer Datei müssen Sie den Schlüsselnamen des Objekts

angeben. Wenn Sie dies nicht tun, API verwendet der den Dateinamen als Schlüsselnamen. Beim Hochladen von Daten aus einem Stream müssen Sie den Schlüsselnamen des Objekts angeben.

Verwenden Sie die Klasse, um erweiterte Upload-Optionen festzulegen, z. B. die Größe des Teils, die Anzahl der Threads beim gleichzeitigen Hochladen der Teile, Metadaten, die Speicherklasse oder. ACL `TransferUtilityUploadRequest`

 Note

Wenn Sie einen Stream für die Datenquelle verwenden, führt die `TransferUtility`-Klasse keine gleichzeitigen Uploads durch.

Im folgenden C#-Beispiel wird eine Datei in mehreren Teilen in einen Amazon-S3-Bucket hochgeladen. Es veranschaulicht, wie Sie zahlreiche `TransferUtility.Upload`-Überladungen zum Hochladen einer Datei verwenden. Jeder nachfolgende Aufruf zum Hochladen ersetzt den vorherigen Upload. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem [AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPUHighLevelAPITest
    {
        private const string bucketName = "**** provide bucket name ****";
        private const string keyName = "**** provide a name for the uploaded object ****";
        private const string filePath = "**** provide the full path name of the file to upload ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
        RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
```

```
    {
        s3Client = new AmazonS3Client(bucketRegion);
        UploadFileAsync().Wait();
    }

private static async Task UploadFileAsync()
{
    try
    {
        var fileTransferUtility =
            new TransferUtility(s3Client);

        // Option 1. Upload a file. The file name is used as the object key
name.

        await fileTransferUtility.UploadAsync(filePath, bucketName);
        Console.WriteLine("Upload 1 completed");

        // Option 2. Specify object key name explicitly.
        await fileTransferUtility.UploadAsync(filePath, bucketName,
keyName);
        Console.WriteLine("Upload 2 completed");

        // Option 3. Upload data from a type of System.IO.Stream.
        using (var fileToUpload =
            new FileStream(filePath, FileMode.Open, FileAccess.Read))
        {
            await fileTransferUtility.UploadAsync(fileToUpload,
                bucketName, keyName);
        }
        Console.WriteLine("Upload 3 completed");

        // Option 4. Specify advanced settings.
        var fileTransferUtilityRequest = new TransferUtilityUploadRequest
        {
            BucketName = bucketName,
            FilePath = filePath,
            StorageClass = S3StorageClass.StandardInfrequentAccess,
            PartSize = 6291456, // 6 MB.
            Key = keyName,
            CannedACL = S3CannedACL.PublicRead
        };
        fileTransferUtilityRequest.Metadata.Add("param1", "Value1");
        fileTransferUtilityRequest.Metadata.Add("param2", "Value2");
    }
}
```

```
        await fileTransferUtility.UploadAsync(fileTransferUtilityRequest);
        Console.WriteLine("Upload 4 completed");
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}
}
```

JavaScript

Example

Laden Sie eine große Datei hoch.

```
import {
    CreateMultipartUploadCommand,
    UploadPartCommand,
    CompleteMultipartUploadCommand,
    AbortMultipartUploadCommand,
    S3Client,
} from "@aws-sdk/client-s3";

const twentyFiveMB = 25 * 1024 * 1024;

export const createString = (size = twentyFiveMB) => {
    return "x".repeat(size);
};

export const main = async () => {
    const s3Client = new S3Client({});
    const bucketName = "test-bucket";
    const key = "multipart.txt";
    const str = createString();
    const buffer = Buffer.from(str, "utf8");
```

```
let uploadId;

try {
  const multipartUpload = await s3Client.send(
    new CreateMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
    }),
  );

  uploadId = multipartUpload.UploadId;

  const uploadPromises = [];
  // Multipart uploads require a minimum size of 5 MB per part.
  const partSize = Math.ceil(buffer.length / 5);

  // Upload each part.
  for (let i = 0; i < 5; i++) {
    const start = i * partSize;
    const end = start + partSize;
    uploadPromises.push(
      s3Client
        .send(
          new UploadPartCommand({
            Bucket: bucketName,
            Key: key,
            UploadId: uploadId,
            Body: buffer.subarray(start, end),
            PartNumber: i + 1,
          }),
        )
        .then((d) => {
          console.log("Part", i + 1, "uploaded");
          return d;
        }),
    );
  }

  const uploadResults = await Promise.all(uploadPromises);

  return await s3Client.send(
    new CompleteMultipartUploadCommand({
      Bucket: bucketName,
```

```

    Key: key,
    UploadId: uploadId,
    MultipartUpload: {
      Parts: uploadResults.map(({ ETag }, i) => ({
        ETag,
        PartNumber: i + 1,
      })),
    },
  })),
);

// Verify the output by downloading the file from the Amazon Simple Storage
Service (Amazon S3) console.
// Because the output is a 25 MB string, text editors might struggle to open the
file.
} catch (err) {
  console.error(err);

  if (uploadId) {
    const abortCommand = new AbortMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
      UploadId: uploadId,
    });

    await s3Client.send(abortCommand);
  }
}
};

```

Example

Laden Sie eine große Datei herunter.

```

import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream } from "fs";

const s3Client = new S3Client({});
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
    Bucket: bucket,
    Key: key,

```

```
    Range: `bytes=${start}-${end}`,
  });

  return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
  const [start, end] = range.split("-");
  return {
    start: parseInt(start),
    end: parseInt(end),
    length: parseInt(length),
  };
};

export const isComplete = ({ end, length }) => end === length - 1;

// When downloading a large file, you might want to break it down into
// smaller pieces. Amazon S3 accepts a Range header to specify the start
// and end of the byte range to be downloaded.
const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    console.log(`Downloading bytes ${nextRange.start} to ${nextRange.end}`);

    const { ContentRange, Body } = await getObjectRange({
      bucket,
      key,
      ...nextRange,
    });

    writeStream.write(await Body.transformToByteArray());
  }
};
```

```
    rangeAndLength = getRangeAndLength(ContentRange);
  }
};

export const main = async () => {
  await downloadInChunks({
    bucket: "my-cool-bucket",
    key: "my-cool-object.txt",
  });
};
```

Go

Example

Laden Sie ein großes Objekt hoch, indem Sie einen Upload-Manager verwenden, um die Daten in Teile zu zerlegen und sie gleichzeitig hochzuladen.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3) actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform bucket
// and object actions.
type BucketBasics struct {
  S3Client *s3.Client
}
```

```
// UploadLargeObject uses an upload manager to upload data to an object in a bucket.
// The upload manager breaks large data into parts and uploads the parts
// concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
  largeObject []byte) error {
  largeBuffer := bytes.NewReader(largeObject)
  var partMiBs int64 = 10
  uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
    u.PartSize = partMiBs * 1024 * 1024
  })
  _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
    Bucket: aws.String(bucketName),
    Key:    aws.String(objectKey),
    Body:   largeBuffer,
  })
}
```



```
    })
    if err != nil {
        log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }

    return err
}
```

Example

Laden Sie ein großes Objekt herunter, indem Sie einen Download-Manager verwenden, um die Daten in Teilen abzurufen und sie gleichzeitig herunterzuladen.

```
// DownloadLargeObject uses a download manager to download an object from a bucket.
// The download manager gets the data in parts and writes them to a buffer until all
// of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey string)
([]byte, error) {
    var partMiBs int64 = 10
    downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader) {
        d.PartSize = partMiBs * 1024 * 1024
    })
    buffer := manager.NewWriteAtBuffer([]byte{})
    _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return buffer.Bytes(), err
}
```

PHP

In diesem Thema wird erklärt, wie Sie die `Aws\S3\Model\MultipartUpload\UploadBuilder` High-Level-Klasse aus dem AWS SDK for PHP für mehrteilige Datei-Uploads.

Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Das folgende PHP Beispiel lädt eine Datei in einen Amazon S3 S3-Bucket hoch. Das Beispiel veranschaulicht, wie Sie Parameter für das `MultipartUploader`-Objekt festlegen.

```
require 'vendor/autoload.php';

use Aws\Exception\MultipartUploadException;
use Aws\S3\MultipartUploader;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Prepare the upload parameters.
$uploader = new MultipartUploader($s3, '/path/to/large/file.zip', [
    'bucket' => $bucket,
    'key'    => $keyname
]);

// Perform the upload.
try {
    $result = $uploader->upload();
    echo "Upload complete: {$result['ObjectURL']}" . PHP_EOL;
} catch (MultipartUploadException $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Python

Im folgenden Beispiel wird ein Objekt mithilfe des mehrteiligen Upload-Pythons auf hoher Ebene API (die `TransferManager` Klasse) geladen.

```
import sys
import threading
```

```
import boto3
from boto3.s3.transfer import TransferConfig

MB = 1024 * 1024
s3 = boto3.resource("s3")

class TransferCallback:
    """
    Handle callbacks from the transfer manager.

    The transfer manager periodically calls the __call__ method throughout
    the upload and download process so that it can take action, such as
    displaying progress to the user and collecting data about the transfer.
    """

    def __init__(self, target_size):
        self._target_size = target_size
        self._total_transferred = 0
        self._lock = threading.Lock()
        self.thread_info = {}

    def __call__(self, bytes_transferred):
        """
        The callback method that is called by the transfer manager.

        Display progress during file transfer and collect per-thread transfer
        data. This method can be called by multiple threads, so shared instance
        data is protected by a thread lock.
        """
        thread = threading.current_thread()
        with self._lock:
            self._total_transferred += bytes_transferred
            if thread.ident not in self.thread_info.keys():
                self.thread_info[thread.ident] = bytes_transferred
            else:
                self.thread_info[thread.ident] += bytes_transferred

            target = self._target_size * MB
            sys.stdout.write(
                f"\r{self._total_transferred} of {target} transferred "
                f"({(self._total_transferred / target) * 100:.2f}%)."
            )
```

```
sys.stdout.flush()
```

```
def upload_with_default_configuration(
    local_file_path, bucket_name, object_key, file_size_mb
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, using the default
    configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Callback=transfer_callback
    )
    return transfer_callback.thread_info

def upload_with_chunksize_and_meta(
    local_file_path, bucket_name, object_key, file_size_mb, metadata=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {"Metadata": metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
        Callback=transfer_callback,
    )
    return transfer_callback.thread_info
```

```
def upload_with_high_threshold(local_file_path, bucket_name, object_key,
    file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info

def upload_with_sse(
    local_file_path, bucket_name, object_key, file_size_mb, sse_key=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, ExtraArgs=extra_args,
        Callback=transfer_callback
    )
    return transfer_callback.thread_info

def download_with_default_configuration(
    bucket_name, object_key, download_file_path, file_size_mb
```

```
):  
    """  
    Download a file from an Amazon S3 bucket to a local folder, using the  
    default configuration.  
    """  
    transfer_callback = TransferCallback(file_size_mb)  
    s3.Bucket(bucket_name).Object(object_key).download_file(  
        download_file_path, Callback=transfer_callback  
    )  
    return transfer_callback.thread_info  
  
def download_with_single_thread(  
    bucket_name, object_key, download_file_path, file_size_mb  
):  
    """  
    Download a file from an Amazon S3 bucket to a local folder, using a  
    single thread.  
    """  
    transfer_callback = TransferCallback(file_size_mb)  
    config = TransferConfig(use_threads=False)  
    s3.Bucket(bucket_name).Object(object_key).download_file(  
        download_file_path, Config=config, Callback=transfer_callback  
    )  
    return transfer_callback.thread_info  
  
def download_with_high_threshold(  
    bucket_name, object_key, download_file_path, file_size_mb  
):  
    """  
    Download a file from an Amazon S3 bucket to a local folder, setting a  
    multipart threshold larger than the size of the file.  
  
    Setting a multipart threshold larger than the size of the file results  
    in the transfer manager sending the file as a standard download instead  
    of a multipart download.  
    """  
    transfer_callback = TransferCallback(file_size_mb)  
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)  
    s3.Bucket(bucket_name).Object(object_key).download_file(  
        download_file_path, Config=config, Callback=transfer_callback  
    )  
    return transfer_callback.thread_info
```

```
def download_with_sse(
    bucket_name, object_key, download_file_path, file_size_mb, sse_key
):
    """
    Download a file from an Amazon S3 bucket to a local folder, adding a
    customer-provided encryption key to the request.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)

    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, ExtraArgs=extra_args, Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

Verwendung der AWS SDKs(niedriges NiveauAPI)

Das Tool AWS SDK macht ein Low-Level verfügbar API, das Amazon S3 REST API für mehrteilige Uploads sehr ähnlich ist (siehe. [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#)). Verwenden Sie das Low-Level API, wenn Sie mehrteilige Uploads anhalten und fortsetzen müssen, die Teilgrößen während des Uploads variieren müssen oder wenn Sie die Größe der Upload-Daten nicht im Voraus kennen. Wenn Sie diese Anforderungen nicht erfüllen, verwenden Sie den High-Level-Modus API (siehe). [Verwendung der AWS SDKs\(auf hohem NiveauAPI\)](#)

Java

Im folgenden Beispiel wird gezeigt, wie Sie mithilfe der Low-Level-Java-Klassen eine Datei hochladen. Es führt die folgenden Schritte aus:

- Startet einen mehrteiligen Upload mit der Methode `AmazonS3Client.initiateMultipartUpload()` und übergibt ein `InitiateMultipartUploadRequest`-Objekt.
- Speichern Sie die Upload-ID, die von der Methode `AmazonS3Client.initiateMultipartUpload()` zurückgegeben wird. Sie geben diese Upload-ID bei jeder nachfolgenden mehrteiligen Upload-Operation an.
- Lädt die Teile des Objekts hoch. Für jedes Teil rufen Sie die Methode `AmazonS3Client.uploadPart()` auf. Sie stellen die Informationen für das Hochladen von Teilen in einem `UploadPartRequest`-Objekt bereit.
- Speichert für jeden Teil die Antwort der `AmazonS3Client.uploadPart()` Methode in einer Liste. ETag Sie verwenden die ETag Werte, um den mehrteiligen Upload abzuschließen.
- Ruft die Methode `AmazonS3Client.completeMultipartUpload()` auf, um den mehrteiligen Upload fertigzustellen.

Example

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class LowLevelMultipartUpload {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";
```



```
String filePath = "*** Path to file to upload ***";

File file = new File(filePath);
long contentLength = file.length();
long partSize = 5 * 1024 * 1024; // Set part size to 5 MB.

try {
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .withRegion(clientRegion)
        .withCredentials(new ProfileCredentialsProvider())
        .build();

    // Create a list of ETag objects. You retrieve ETags for each object
part
    // uploaded,
    // then, after each individual part has been uploaded, pass the list of
ETags to
    // the request to complete the upload.
    List<PartETag> partETags = new ArrayList<PartETag>();

    // Initiate the multipart upload.
    InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(bucketName, keyName);
    InitiateMultipartUploadResult initResponse =
s3Client.initiateMultipartUpload(initRequest);

    // Upload the file parts.
    long filePosition = 0;
    for (int i = 1; filePosition < contentLength; i++) {
        // Because the last part could be less than 5 MB, adjust the part
size as
        // needed.
        partSize = Math.min(partSize, (contentLength - filePosition));

        // Create the request to upload a part.
        UploadPartRequest uploadRequest = new UploadPartRequest()
            .withBucketName(bucketName)
            .withKey(keyName)
            .withUploadId(initResponse.getUploadId())
            .withPartNumber(i)
            .withFileOffset(filePosition)
            .withFile(file)
            .withPartSize(partSize);
```

```
        // Upload the part and add the response's ETag to our list.
        UploadPartResult uploadResult = s3Client.uploadPart(uploadRequest);
        partETags.add(uploadResult.getPartETag());

        filePosition += partSize;
    }

    // Complete the multipart upload.
    CompleteMultipartUploadRequest compRequest = new
CompleteMultipartUploadRequest(bucketName, keyName,
        initResponse.getUploadId(), partETags);
    s3Client.completeMultipartUpload(compRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

.NET

Das folgende C#-Beispiel zeigt, wie das Low-Level verwendet wird AWS SDK for .NET mehrteiliger UploadAPI, um eine Datei in einen S3-Bucket hochzuladen. Weitere Informationen über mehrteilige Uploads zu Amazon S3 finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Note

Wenn Sie den verwenden AWS SDK for .NET API Beim Hochladen großer Objekte kann es zu einem Timeout kommen, während Daten in den Anforderungsstream geschrieben werden. Sie können mit `UploadPartRequest` ein explizites Timeout festlegen.

Im folgenden C#-Beispiel wird eine Datei mithilfe des mehrteiligen Low-Level-Uploads in einen S3-Bucket hochgeladen. API Informationen zum Einrichten und Ausführen der Codebeispiele

finden [Sie unter Getting Started with the AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPULowLevelAPITest
    {
        private const string bucketName = "**** provide bucket name ****";
        private const string keyName = "**** provide a name for the uploaded object ****";
        private const string filePath = "**** provide the full path name of the file to upload ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Uploading an object");
            UploadObjectAsync().Wait();
        }

        private static async Task UploadObjectAsync()
        {
            // Create list to store upload part responses.
            List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

            // Setup information required to initiate the multipart upload.
            InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
            {
```

```
        BucketName = bucketName,
        Key = keyName
    };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initWithRequest);

    // Upload parts.
    long contentLength = new FileInfo(filePath).Length;
    long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

    try
    {
        Console.WriteLine("Uploading parts");

        long filePosition = 0;
        for (int i = 1; filePosition < contentLength; i++)
        {
            UploadPartRequest uploadRequest = new UploadPartRequest
            {
                BucketName = bucketName,
                Key = keyName,
                UploadId = initResponse.UploadId,
                PartNumber = i,
                PartSize = partSize,
                FilePosition = filePosition,
                FilePath = filePath
            };

            // Track upload progress.
            uploadRequest.StreamTransferProgress +=
                new
                EventHandler<StreamTransferProgressArgs>(UploadPartProgressEventCallback);

            // Upload a part and add the response to our list.
            uploadResponses.Add(await
                s3Client.UploadPartAsync(uploadRequest));

            filePosition += partSize;
        }

        // Setup to complete the upload.
```

```
        CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
    {
        BucketName = bucketName,
        Key = keyName,
        UploadId = initResponse.UploadId
    };
    completeRequest.AddPartETags(uploadResponses);

    // Complete the upload.
    CompleteMultipartUploadResponse completeUploadResponse =
        await s3Client.CompleteMultipartUploadAsync(completeRequest);
    }
    catch (Exception exception)
    {
        Console.WriteLine("An AmazonS3Exception was thrown: { 0}",
exception.Message);

        // Abort the upload.
        AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
    {
        BucketName = bucketName,
        Key = keyName,
        UploadId = initResponse.UploadId
    };
        await s3Client.AbortMultipartUploadAsync(abortMPURequest);
    }
    }
    public static void UploadPartProgressEventCallback(object sender,
StreamTransferProgressArgs e)
    {
        // Process event.
        Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
    }
    }
}
```

PHP

Dieses Thema zeigt, wie Sie die `uploadPart` Low-Level-Methode aus Version 3 von verwenden AWS SDK for PHP um eine Datei in mehreren Teilen hochzuladen. Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Das folgende PHP Beispiel lädt eine Datei mithilfe des PHP API mehrteiligen Low-Level-Uploads in einen Amazon S3 S3-Bucket hoch.

```
require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$filename = '*** Path to and Name of the File to Upload ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

$result = $s3->createMultipartUpload([
    'Bucket' => $bucket,
    'Key' => $keyname,
    'StorageClass' => 'REDUCED_REDUNDANCY',
    'Metadata' => [
        'param1' => 'value 1',
        'param2' => 'value 2',
        'param3' => 'value 3'
    ]
]);

$uploadId = $result['UploadId'];

// Upload the file in parts.
try {
    $file = fopen($filename, 'r');
    $partNumber = 1;
    while (!feof($file)) {
        $result = $s3->uploadPart([
            'Bucket' => $bucket,
            'Key' => $keyname,
            'UploadId' => $uploadId,
            'PartNumber' => $partNumber,
            'Body' => fread($file, 5 * 1024 * 1024),
        ]);
        $parts['Parts'][$partNumber] = [
            'PartNumber' => $partNumber,
```

```
        'ETag' => $result['ETag'],
    ];
    $partNumber++;

    echo "Uploading part $partNumber of $filename." . PHP_EOL;
}
fclose($file);
} catch (S3Exception $e) {
    $result = $s3->abortMultipartUpload([
        'Bucket'    => $bucket,
        'Key'       => $keyname,
        'UploadId' => $uploadId
    ]);

    echo "Upload of $filename failed." . PHP_EOL;
}

// Complete the multipart upload.
$result = $s3->completeMultipartUpload([
    'Bucket'    => $bucket,
    'Key'       => $keyname,
    'UploadId' => $uploadId,
    'MultipartUpload' => $parts,
]);
$url = $result['Location'];

echo "Uploaded $filename to $url." . PHP_EOL;
```

Verwendung der AWS SDK for Ruby

Das Tool AWS SDK for Ruby Version 3 unterstützt mehrteilige Amazon S3 S3-Uploads auf zwei Arten. Bei der ersten Option können Sie verwaltete Datei-Uploads verwenden. Weitere Informationen finden Sie unter [Hochladen von Dateien auf Amazon S3](#) in der AWS Blog für Entwickler. Verwaltete Datei-Uploads sind die empfohlene Methode zum Hochladen von Dateien zu einem Bucket. Sie bieten die folgenden Vorteile:

- Verwaltet mehrteilige Uploads von Objekten über 15 MB.
- Dateien werden im Binärmodus korrekt geöffnet, um Codierungsprobleme zu vermeiden.
- Verwendet mehrere Threads zum parallelen Hochladen von Teilen großer Objekte.

Alternativ können Sie die folgenden Multipart-Upload-Client-Vorgänge direkt verwenden:

- [create_multipart_upload](#) – Initiiert einen mehrteiligen Upload und gibt eine Upload-ID zurück.
- [upload_part](#) – Lädt einen Teil eines mehrteiligen Uploads hoch.
- [upload_part_copy](#) – Lädt einen Teil durch Kopieren von Daten aus einem vorhandenen Objekt hoch, das als Datenquelle dient.
- [complete_multipart_upload](#) – Schließt einen mehrteiligen Upload durch das Zusammenfügen zuvor hochgeladener Teile ab.
- [abort_multipart_upload](#) – Bricht einen mehrteiligen Upload ab.

Laden eines Verzeichnisses mit der High-Level-Version hoch. NET TransferUtility Klasse

Mit der Klasse `TransferUtility` können Sie ein gesamtes Verzeichnis hochladen. Standardmäßig werden nur die Dateien im Stammverzeichnis des angegebenen Verzeichnisses API hochgeladen. Sie können jedoch festlegen, dass die Dateien in allen Unterverzeichnissen rekursiv hochgeladen werden sollen.

Um Dateien im angegebenen Verzeichnis basierend auf Filterkriterien anzugeben, geben Sie Filterausdrücke an. Wenn Sie z. B. nur die `.pdf`-Dateien aus einem Verzeichnis hochladen möchten, geben Sie den Filterausdruck `"* .pdf"` ein.

Wenn Sie Dateien aus einem Verzeichnis hochladen, geben Sie nicht die Schlüsselnamen der resultierenden Objekte an. Amazon S3 setzt die Schlüsselnamen aus dem Original-Dateipfad zusammen. Angenommen, ein Verzeichnis mit dem Namen `c:\myfolder` besitzt die folgende Verzeichnisstruktur:

Example

```
C:\myfolder
  \a.txt
  \b.pdf
  \media\
    An.mp3
```

Wenn Sie dieses Verzeichnis hochladen, verwendet Amazon S3 die folgenden Schlüsselnamen:

Example

```
a.txt
b.pdf
media/An.mp3
```

Example

Im folgenden C#-Beispiel wird ein Verzeichnis in einen Amazon-S3-Bucket hochgeladen. Es veranschaulicht, wie Sie zahlreiche `TransferUtility.UploadDirectory`-Überladungen zum Hochladen des Verzeichnisses verwenden. Jeder nachfolgende Aufruf zum Hochladen ersetzt den vorherigen Upload. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadDirMPUHighLevelAPITest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string directoryPath = @"*** directory path ***";
        // The example uploads only .txt files.
        private const string wildcard = "*.txt";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadDirAsync().Wait();
        }

        private static async Task UploadDirAsync()
        {
            try
            {
```

```
var directoryTransferUtility =
    new TransferUtility(s3Client);

// 1. Upload a directory.
await directoryTransferUtility.UploadDirectoryAsync(directoryPath,
    existingBucketName);
Console.WriteLine("Upload statement 1 completed");

// 2. Upload only the .txt files from a directory
//    and search recursively.
await directoryTransferUtility.UploadDirectoryAsync(
    directoryPath,
    existingBucketName,
    wildCard,
    SearchOption.AllDirectories);
Console.WriteLine("Upload statement 2 completed");

// 3. The same as Step 2 and some optional configuration.
//    Search recursively for .txt files to upload.
var request = new TransferUtilityUploadDirectoryRequest
{
    BucketName = existingBucketName,
    Directory = directoryPath,
    SearchOption = SearchOption.AllDirectories,
    SearchPattern = wildCard
};

await directoryTransferUtility.UploadDirectoryAsync(request);
Console.WriteLine("Upload statement 3 completed");
}
catch (AmazonS3Exception e)
{
    Console.WriteLine(
        "Error encountered ***. Message:'{0}' when writing an object",
e.Message);
}
catch (Exception e)
{
    Console.WriteLine(
        "Unknown encountered on server. Message:'{0}' when writing an
object", e.Message);
}
}
```

}

Auflisten von mehrteiligen Uploads

Sie können das AWS SDKs(Low-LevelAPI), um eine Liste der laufenden mehrteiligen Uploads in Amazon S3 abzurufen.

Auflisten von mehrteiligen Uploads mit dem AWS SDK(niedriges Niveau) API

Java

Die folgenden Aufgaben führen Sie durch die Verwendung von Java Low-Level-Klassen, um alle laufenden mehrteiligen Uploads für einen Bucket aufzulisten.

Prozess zum Auflisten API mehrteiliger Uploads auf niedriger Ebene

- 1 Erstellen Sie eine Instance der `ListMultipartUploadsRequest` -Klasse und stellen Sie den Bucket-Namen bereit.
- 2 Führen Sie die `AmazonS3Client.listMultipartUploads` -Methode aus. Die Methode gibt eine Instance der Klasse `MultipartUploadListing` zurück, die Ihnen Informationen über die laufenden mehrteiligen Uploads bereitstellt.

Im folgenden Java-Codebeispiel werden die vorherigen Aufgaben veranschaulicht.

Example

```
ListMultipartUploadsRequest allMultipartUploadsRequest =  
    new ListMultipartUploadsRequest(existingBucketName);  
MultipartUploadListing multipartUploadListing =  
    s3Client.listMultipartUploads(allMultipartUploadsRequest);
```

.NET

Um alle laufenden mehrteiligen Uploads in einem bestimmten Bucket aufzulisten, verwenden Sie den AWS SDK for .NET Klasse für mehrteilige Uploads auf niedriger Ebene. API `ListMultipartUploadsRequest` Die Methode `AmazonS3Client.ListMultipartUploads` gibt eine Instance der Klasse `ListMultipartUploadsResponse` zurück, die Informationen über die laufenden mehrteiligen Uploads bereitstellt.

Ein laufender mehrteiliger Upload ist ein mehrteiliger Upload, der von der Anfrage für das Initiieren mehrteiliger Uploads initiiert, aber weder abgeschlossen noch abgebrochen wurde. Weitere Informationen über mehrteilige Amazon-S3-Uploads finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Das folgende C#-Beispiel zeigt, wie Sie den AWS SDK for .NET um alle laufenden mehrteiligen Uploads in einem Bucket aufzulisten. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest
{
    BucketName = bucketName // Bucket receiving the uploads.
};

ListMultipartUploadsResponse response = await
    AmazonS3Client.ListMultipartUploadsAsync(request);
```

PHP

Dieses Thema zeigt, wie Sie die API Low-Level-Klassen aus Version 3 von verwenden AWS SDK for PHP um alle laufenden mehrteiligen Uploads in einem Bucket aufzulisten. Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Das folgende PHP Beispiel zeigt die Auflistung aller laufenden mehrteiligen Uploads in einem Bucket.

```
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Retrieve a list of the current multipart uploads.
$result = $s3->listMultipartUploads([
    'Bucket' => $bucket
]);
```

```
// Write the list of uploads to the page.  
print_r($result->toArray());
```

Auflisten von mehrteiligen Uploads mit dem REST API

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird beschrieben, wie Sie mehrteilige Uploads auflisten können: REST API

- [ListParts](#)-listet die hochgeladenen Teile für einen bestimmten mehrteiligen Upload auf.
- [ListMultipartUploads](#)-listet laufende mehrteilige Uploads auf.

Auflisten von mehrteiligen Uploads mit dem AWS CLI

Die folgenden Abschnitte in der AWS Command Line Interface beschreibt die Operationen zum Auflisten mehrteiliger Uploads.

- [list-parts](#) – listet die hochgeladenen Teile für einen bestimmten mehrteiligen Upload auf.
- [list-multipart-uploads](#)-listet laufende mehrteilige Uploads auf.

Verfolgen eines mehrteiligen Uploads

Der mehrteilige Upload auf hoher Ebene API bietet eine Listen-Schnittstelle `ProgressListener`, um den Upload-Fortschritt beim Hochladen eines Objekts auf Amazon S3 zu verfolgen. Fortschrittsereignisse treten periodisch auf und benachrichtigen den Listener, dass Bytes übertragen wurden.

Java

Example

```
TransferManager tm = new TransferManager(new ProfileCredentialsProvider());  
  
PutObjectRequest request = new PutObjectRequest(  
    existingBucketName, keyName, new File(filePath));  
  
// Subscribe to the event and provide event handler.  
request.setProgressListener(new ProgressListener() {
```

```
public void progressChanged(ProgressEvent event) {
    System.out.println("Transferred bytes: " +
        event.getBytesTransferred());
}
});
```

Example

Der folgende Java-Code lädt eine Datei hoch und verwendet den `ProgressListener`, um den Upload-Fortschritt zu verfolgen. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import java.io.File;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.event.ProgressEvent;
import com.amazonaws.event.ProgressListener;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.Upload;

public class TrackMPUPProgressUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "*** Provide bucket name ***";
        String keyName            = "*** Provide object key ***";
        String filePath           = "*** file to upload ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

        // For more advanced uploads, you can create a request object
        // and supply additional request parameters (ex: progress listeners,
        // canned ACLs, etc.)
        PutObjectRequest request = new PutObjectRequest(
            existingBucketName, keyName, new File(filePath));

        // You can ask the upload for its progress, or you can
        // add a ProgressListener to your request to receive notifications
        // when bytes are transferred.
        request.setGeneralProgressListener(new ProgressListener() {
            @Override
```

```
public void progressChanged(ProgressEvent progressEvent) {
    System.out.println("Transferred bytes: " +
        progressEvent.getBytesTransferred());
}
});

// TransferManager processes all transfers asynchronously,
// so this call will return immediately.
Upload upload = tm.upload(request);

try {
    // You can block and wait for the upload to finish
    upload.waitForCompletion();
} catch (AmazonClientException amazonClientException) {
    System.out.println("Unable to upload file, upload aborted.");
    amazonClientException.printStackTrace();
}
}
```

.NET

Das folgende C#-Beispiel lädt eine Datei unter Verwendung der Klasse `TransferUtility` in einen S3-Bucket hoch und verfolgt den Fortschritt des Uploads. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TrackMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide the bucket name ***";
        private const string keyName = "*** provide the name for the uploaded object ***";
        private const string filePath = " *** provide the full path name of the file to upload ***";
        // Specify your bucket region (an example region is shown).
```

```
private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
private static IAmazonS3 s3Client;

public static void Main()
{
    s3Client = new AmazonS3Client(bucketRegion);
    TrackMPUAsync().Wait();
}

private static async Task TrackMPUAsync()
{
    try
    {
        var fileTransferUtility = new TransferUtility(s3Client);

        // Use TransferUtilityUploadRequest to configure options.
        // In this example we subscribe to an event.
        var uploadRequest =
            new TransferUtilityUploadRequest
            {
                BucketName = bucketName,
                FilePath = filePath,
                Key = keyName
            };

        uploadRequest.UploadProgressEvent +=
            new EventHandler<UploadProgressArgs>
                (uploadRequest_UploadPartProgressEvent);

        await fileTransferUtility.UploadAsync(uploadRequest);
        Console.WriteLine("Upload completed");
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}
```



```
    }

    static void uploadRequest_UploadPartProgressEvent(object sender,
UploadProgressArgs e)
    {
        // Process event.
        Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
    }
}
}
```

Abbrechen eines mehrteiligen Uploads

Nachdem Sie einen mehrteiligen Upload initiiert haben, beginnen Sie mit dem Hochladen von Teilen. Amazon S3 speichert diese Teile, aber das Objekt wird erst dann aus den Teilen erstellt, nachdem Sie alle Teile hochgeladen und eine `successful`-Anfrage gesendet haben, um den mehrteiligen Upload abzuschließen (Sie sollten sicherstellen, dass Ihre Anfrage auf Abschluss des mehrteiligen Uploads erfolgreich ist). Nach dem Empfang der Anfrage auf Abschluss des Multipart-Uploads erzeugt Amazon S3 aus den Teilen ein Objekt. Wenn die Anforderung für den Abschluss des mehrteiligen Uploads nicht erfolgreich ist, fügt Amazon S3 die Teile nicht zusammen und erstellt daher auch kein Objekt.

Ihnen wird der gesamte Speicher in Rechnung gestellt, der mit hochgeladenen Teilen verknüpft ist. Weitere Informationen finden Sie unter [Mehrteiliger Upload und Preise](#). Deshalb sollten Sie den mehrteiligen Upload abschließen, damit das Objekt erstellt wird, oder den Multipart-Upload abbrechen, um hochgeladene Teile zu entfernen.

Sie können einen laufenden mehrteiligen Upload in Amazon S3 beenden, indem Sie AWS Command Line Interface (AWS CLI), oder REST API AWS SDKs. Sie können einen unvollständigen mehrteiligen Upload auch mithilfe einer Bucket-Lebenszykluskonfiguration beenden.

Verwendung der AWS SDKs(auf hohem NiveauAPI)

Java

Die Klasse `TransferManager` bietet die Methode `abortMultipartUploads`, um laufende Multipart-Uploads zu stoppen. Ein Upload wird als laufend betrachtet, nachdem Sie ihn initiiert haben und bis er abgeschlossen ist oder Sie ihn abbrechen. Sie geben einen `Date` Wert an, und dadurch werden alle mehrteiligen Uploads für diesen Bucket API gestoppt, die vor dem angegebenen Wert initiiert wurden `Date` und noch im Gange sind.

Die folgenden Aufgaben führen Sie durch die Verwendung von High-Level Java-Klassen für das Abbrechen mehrteiliger Uploads.

Prozess zum Stoppen API mehrteiliger Uploads auf hoher Ebene

- 1 Erstellen Sie eine Instance der `TransferManager` -Klasse.
- 2 Führen Sie die `TransferManager.abortMultipartUploads` -Methode aus, indem Sie den Bucket-Namen und einen Date-Wert übergeben.

Der folgende Java-Code bricht alle laufenden mehrteiligen Uploads ab, die vor mehr als einer Woche für einen bestimmten Bucket initiiert wurden. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import java.util.Date;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.transfer.TransferManager;

public class AbortMPUUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "**** Provide existing bucket name ****";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

        int sevenDays = 1000 * 60 * 60 * 24 * 7;
        Date oneWeekAgo = new Date(System.currentTimeMillis() - sevenDays);

        try {
            tm.abortMultipartUploads(existingBucketName, oneWeekAgo);
        } catch (AmazonClientException amazonClientException) {
            System.out.println("Unable to upload file, upload was aborted.");
            amazonClientException.printStackTrace();
        }
    }
}
```

Note

Sie können auch einen bestimmten mehrteiligen Upload abbrechen. Weitere Informationen finden Sie unter [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#).

.NET

Das folgende C#-Beispiel bricht alle laufenden mehrteiligen Uploads ab, die vor mehr als einer Woche für einen bestimmten Bucket initiiert wurden. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class AbortMPUUsingHighLevelAPITest
    {
        private const string bucketName = "**** provide bucket name ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            AbortMPUAsync().Wait();
        }

        private static async Task AbortMPUAsync()
        {
            try
            {
                var transferUtility = new TransferUtility(s3Client);
```

```
        // Abort all in-progress uploads initiated before the specified
date.
        await transferUtility.AbortMultipartUploadsAsync(
            bucketName, DateTime.Now.AddDays(-7));
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}
}
```

Note

Sie können auch einen bestimmten mehrteiligen Upload abbrechen. Weitere Informationen finden Sie unter [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#).

Verwendung der AWS SDKs(niedriges NiveauAPI)

Durch Aufruf der Methode `AmazonS3.abortMultipartUpload` können Sie einen mehrteiligen Upload in Bearbeitung abbrechen. Diese Methode löscht alle Teile, die in Amazon S3 hochgeladen wurden, und gibt die Ressourcen frei. Sie müssen die Upload-ID, den Bucket-Namen und die Schlüsselnamen bereitstellen. Das folgende Java-Codebeispiel zeigt, wie ein laufender mehrteiliger Upload abgebrochen wird.

Um einen mehrteiligen Upload abzubrechen, stellen Sie die Upload-ID bereit sowie die Bucket- und Schlüsselnamen, die im Upload verwendet werden. Nachdem ein mehrteiliger Upload abgebrochen wurde, können Sie mit dieser Upload-ID keinen zusätzlichen Teile mehr hochladen. Weitere Informationen über mehrteilige Amazon-S3-Uploads finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Java

Im folgenden Java-Codebeispiel wird ein laufender mehrteiliger Upload gestoppt.

Example

```
InitiateMultipartUploadRequest initRequest =
    new InitiateMultipartUploadRequest(existingBucketName, keyName);
InitiateMultipartUploadResult initResponse =
    s3Client.initiateMultipartUpload(initRequest);

AmazonS3 s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
s3Client.abortMultipartUpload(new AbortMultipartUploadRequest(
    existingBucketName, keyName, initResponse.getUploadId()));
```

Note

Statt einen spezifischen mehrteiligen Upload können Sie auch alle Ihre mehrteiligen Uploads abbrechen, die vor einem bestimmten Zeitpunkt initiiert wurden und immer noch laufen. Diese Bereinigungsoperation ist praktisch, um alte mehrteilige Uploads abzubrechen, die Sie initiiert haben, aber die nicht abgeschlossen oder abgebrochen wurden. Weitere Informationen finden Sie unter [Verwendung der AWS SDKs\(auf hohem NiveauAPI\)](#).

.NET

Das folgende C#-Beispiel veranschaulicht, wie Sie einen mehrteiligen Upload abbrechen. Ein vollständiges C#-Beispiel mit dem folgenden Code finden Sie unter [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#).

```
AbortMultipartUploadRequest abortMPURequest = new AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = keyName,
    UploadId = initResponse.UploadId
};
await AmazonS3Client.AbortMultipartUploadAsync(abortMPURequest);
```

Sie können auch alle laufenden mehrteiligen Uploads abbrechen, die vor einem bestimmten Zeitpunkt initiiert wurden. Diese Bereinigungsoperation ist praktisch, um mehrteilige Uploads

abzubrechen, die nicht abgeschlossen oder abgebrochen wurden. Weitere Informationen finden Sie unter [Verwendung der AWS SDKs\(auf hohem NiveauAPI\)](#).

PHP

Dieses Beispiel zeigt, wie Sie eine Klasse aus Version 3 von verwenden AWS SDK for PHP um einen laufenden mehrteiligen Upload abzubrechen. Für weitere Informationen über AWS SDKfür Ruby API gehen Sie zu [AWS SDKfür Ruby - Version 2](#). Das Beispiel der Methode `abortMultipartUpload()`.

Für weitere Informationen über AWS SDKfür Ruby API gehen Sie zu [AWS SDKfür Ruby - Version 2](#).

```
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$uploadId = '*** Upload ID of upload to Abort ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Abort the multipart upload.
$s3->abortMultipartUpload([
    'Bucket'   => $bucket,
    'Key'      => $keyname,
    'UploadId' => $uploadId,
]);
```

Mit dem REST API

Weitere Informationen zur Verwendung von REST API zum Stoppen eines mehrteiligen Uploads finden Sie [AbortMultipartUpload](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS CLI

Weitere Informationen zur Verwendung von AWS CLI Informationen zum Beenden eines mehrteiligen Uploads finden Sie [abort-multipart-upload](#) in der AWS CLI Befehlsreferenz.

Kopieren eines Objekts mit Multipart-Upload

Die Beispiele in diesem Abschnitt zeigen Ihnen, wie Sie Objekte mit mehr als 5 GB mithilfe des mehrteiligen Uploads kopieren können. API Sie können in einer einzelnen Operation Objekte bis zu einer Größe von 5 GB hochladen. Weitere Informationen finden Sie unter [Objekte kopieren, verschieben und umbenennen](#).

Verwendung der AWS SDKs

Gehen Sie wie folgt vor, um ein Objekt mithilfe der API Low-Level-Methode zu kopieren:

- Initiieren eines mehrteiligen Uploads durch Aufrufen der Methode `AmazonS3Client.initiateMultipartUpload()`.
- Speichern Sie die Upload-ID aus dem Antwortobjekt, das die Methode `AmazonS3Client.initiateMultipartUpload()` zurückgibt. Sie geben diese Upload-ID bei jeder Teiloperation mehrteiliger Uploads an.
- Kopieren Sie alle Teile. Erstellen Sie für jeden Teil, den Sie kopieren müssen, eine neue Instance der Klasse `CopyPartRequest`. Geben Sie die teilspezifischen Informationen an, darunter Quell- und Zielbucket-Namen, Quell- und Ziel-Objektschlüssel, Upload-ID, Stelle des ersten und des letzten Byte des Teils sowie die Nummer des Teils.
- Speichern Sie die Antworten der `AmazonS3Client.copyPart()`-Methodenaufrufe. Jede Antwort enthält den Wert für ETag und die Teilenummer des hochgeladenen Teils. Sie benötigen diese Informationen, um den mehrteiligen Upload abzuschließen.
- Rufen Sie die Methode `AmazonS3Client.completeMultipartUpload()` auf, um die Kopieroperation abzuschließen.

Java

Example

Das folgende Beispiel zeigt, wie Sie Amazon S3 Low-Level-Java verwendenAPI, um eine mehrteilige Kopie durchzuführen. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class LowLevelMultipartCopy {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String sourceBucketName = "**** Source bucket name ****";
        String sourceObjectKey = "**** Source object key ****";
        String destBucketName = "**** Target bucket name ****";
        String destObjectKey = "**** Target object key ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(destBucketName,
                                destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(sourceBucketName, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
```



```
        // The last part might be smaller than partSize, so check to make
sure
        // that lastByte isn't beyond the end of the object.
        long lastByte = Math.min(bytePosition + partSize - 1, objectSize -
1);

        // Copy this part.
        CopyPartRequest copyRequest = new CopyPartRequest()
            .withSourceBucketName(sourceBucketName)
            .withSourceKey(sourceObjectKey)
            .withDestinationBucketName(destBucketName)
            .withDestinationKey(destObjectKey)
            .withUploadId(initResult.getUploadId())
            .withFirstByte(bytePosition)
            .withLastByte(lastByte)
            .withPartNumber(partNum++);
        copyResponses.add(s3Client.copyPart(copyRequest));
        bytePosition += partSize;
    }

    // Complete the upload request to concatenate all uploaded parts and
make the
    // copied object available.
    CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
        destBucketName,
        destObjectKey,
        initResult.getUploadId(),
        getETags(copyResponses));
    s3Client.completeMultipartUpload(completeRequest);
    System.out.println("Multipart copy complete.");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

// This is a helper function to construct a list of ETags.
private static List<PartETag> getETags(List<CopyPartResult> responses) {
```

```
List<PartETag> etags = new ArrayList<PartETag>();
for (CopyPartResult response : responses) {
    etags.add(new PartETag(response.getPartNumber(), response.getETag()));
}
return etags;
}
}
```

.NET

Das folgende C#-Beispiel zeigt die Verwendung von AWS SDK for .NET um ein Amazon S3 S3-Objekt, das größer als 5 GB ist, von einem Quellspeicherort an einen anderen zu kopieren, z. B. von einem Bucket in einen anderen. Um Objekte zu kopieren, die kleiner als 5 GB sind, verwenden Sie das Kopierverfahren in einer einzigen Operation, das unter [Verwendung der AWS SDKs](#) beschrieben wird. Weitere Informationen über mehrteilige Amazon-S3-Uploads finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Dieses Beispiel zeigt, wie Sie ein Amazon S3 S3-Objekt, das größer als 5 GB ist, von einem S3-Bucket in einen anderen kopieren, indem Sie den AWS SDK for .NET mehrteiliger UploadAPI.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectUsingMPUapiTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
source object ***";
        private const string targetBucket = "*** provide the name of the bucket to
copy the object to ***";
        private const string sourceObjectKey = "*** provide the name of object to
copy ***";
        private const string targetObjectKey = "*** provide the name of the object
copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
```

```
private static IAmazonS3 s3Client;

public static void Main()
{
    s3Client = new AmazonS3Client(bucketRegion);
    Console.WriteLine("Copying an object");
    MPUCopyObjectAsync().Wait();
}
private static async Task MPUCopyObjectAsync()
{
    // Create a list to store the upload part responses.
    List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();
    List<CopyPartResponse> copyResponses = new List<CopyPartResponse>();

    // Setup information required to initiate the multipart upload.
    InitiateMultipartUploadRequest initiateRequest =
        new InitiateMultipartUploadRequest
        {
            BucketName = targetBucket,
            Key = targetObjectKey
        };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // Save the upload ID.
    String uploadId = initResponse.UploadId;

    try
    {
        // Get the size of the object.
        GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
        {
            BucketName = sourceBucket,
            Key = sourceObjectKey
        };

        GetObjectMetadataResponse metadataResponse =
            await s3Client.GetObjectMetadataAsync(metadataRequest);
        long objectSize = metadataResponse.ContentLength; // Length in
bytes.
```

```
// Copy the parts.
long partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

long bytePosition = 0;
for (int i = 1; bytePosition < objectSize; i++)
{
    CopyPartRequest copyRequest = new CopyPartRequest
    {
        DestinationBucket = targetBucket,
        DestinationKey = targetObjectKey,
        SourceBucket = sourceBucket,
        SourceKey = sourceObjectKey,
        UploadId = uploadId,
        FirstByte = bytePosition,
        LastByte = bytePosition + partSize - 1 >= objectSize ?
objectSize - 1 : bytePosition + partSize - 1,
        PartNumber = i
    };

    copyResponses.Add(await s3Client.CopyPartAsync(copyRequest));

    bytePosition += partSize;
}

// Set up to complete the copy.
CompleteMultipartUploadRequest completeRequest =
new CompleteMultipartUploadRequest
{
    BucketName = targetBucket,
    Key = targetObjectKey,
    UploadId = initResponse.UploadId
};
completeRequest.AddPartETags(copyResponses);

// Complete the copy.
CompleteMultipartUploadResponse completeUploadResponse =
    await s3Client.CompleteMultipartUploadAsync(completeRequest);
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
}
```

```
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
    }
}
```

Mit dem REST API

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird das REST API für mehrteilige Uploads beschrieben. Um ein vorhandenes Objekt zu kopieren, verwenden Sie den Upload Part (Copy) API und geben Sie das Quellobjekt an, indem Sie Ihrer `x-amz-copy-source` Anfrage den Anforderungsheader hinzufügen.

- [Initiieren eines mehrteiligen Uploads](#)
- [Upload Part](#)
- [Teil hochladen \(kopieren\)](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Abort Multipart Upload](#)
- [Teile auflisten](#)
- [List Multipart Uploads](#)

Sie können diese verwenden APIs, um Ihre eigenen REST Anfragen zu stellen, oder Sie können eine der von SDKs uns bereitgestellten verwenden. Weitere Informationen zur Verwendung von Multipart Upload mit dem AWS CLI, finden Sie unter [Verwendung der AWS CLI](#). Weitere Informationen zu dem finden Sie SDKs unter [AWS SDK-Unterstützung für mehrteilige Uploads](#).

Einschränkungen mehrteiliger Amazon-S3-Uploads

Die folgende Tabelle enthält die Core-Spezifikationen für den mehrteiligen Upload. Weitere Informationen finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Item	Spezifikation
Maximale Objektgröße	5 TiB

Item	Spezifikation
Maximale Anzahl von Teilen pro Upload	10.000
Teilenummern	1 bis 10.000 (inklusive)
Teilegröße	5 MiB bis 5 GiB. Es gibt keine minimale Größenbeschränkung für den letzten Teil Ihres mehrteiligen Uploads.
Maximale Anzahl der zurückgegebenen Teile bei einer Anforderung zum Auflisten der Teile	1000
Maximale Anzahl der zurückgegebenen mehrteiligen Uploads bei einer Anforderung zum Auflisten mehrteiliger Uploads	1000

Fügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu

Sie können bedingte Anfragen verwenden, um Ihren S3-Vorgängen Vorbedingungen hinzuzufügen. Um bedingte Anfragen zu verwenden, fügen Sie Ihrem Amazon S3 API S3-Vorgang einen zusätzlichen Header hinzu. Dieser Header spezifiziert eine Bedingung, die, wenn sie nicht erfüllt wird, dazu führt, dass der S3-Vorgang fehlschlägt.

Bedingte Lesevorgänge werden für GETHEAD, und COPY Anfragen unterstützt. Sie können Vorbedingungen hinzufügen, um ein Objekt auf der Grundlage seines Entitäts-Tags (ETag) oder des Datums der letzten Änderung zurückzugeben oder zu kopieren. Dadurch kann ein S3-Vorgang auf Objekte beschränkt werden, die seit einem bestimmten Datum aktualisiert wurden. Sie können einen S3-Vorgang auch auf einen bestimmten Wert beschränken ETag. Dadurch könnte sichergestellt werden, dass Sie nur eine bestimmte Objektversion zurückgeben oder kopieren. Weitere Informationen zu Objekt-Metadaten erhalten Sie unter [Arbeiten mit Objekt-Metadaten](#).

Durch bedingte Schreibvorgänge kann sichergestellt werden, dass während des PUT Betriebs kein Objekt mit demselben Schlüsselnamen in Ihrem Bucket vorhanden ist. Dadurch wird verhindert,

dass vorhandene Objekte mit identischen Schlüsselnamen überschrieben werden. Sie können bedingte Schreibvorgänge verwenden für [PutObject](#) oder [CompleteMultipartUpload](#) Anfragen. Weitere Informationen zu Schlüsselnamen finden Sie unter [Erstellen von Objektschlüsselnamen](#).

Für bedingte Anfragen fallen keine zusätzlichen Gebühren an. Ihnen werden nur die bestehenden Tarife für die entsprechenden Anfragen berechnet, auch für fehlgeschlagene Anfragen. Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon-S3-Preise](#).

Themen

- [Wie ruft man Objekte auf der Grundlage von Metadaten mit bedingten Lesevorgängen ab oder kopiert sie](#)
- [Wie verhindert man Objektüberschreibungen durch bedingte Schreibvorgänge](#)

Wie ruft man Objekte auf der Grundlage von Metadaten mit bedingten Lesevorgängen ab oder kopiert sie

Bei bedingten Lesevorgängen können Sie Ihrer Leseanforderung einen zusätzlichen Header hinzufügen, um Ihrem S3-Vorgang Vorbedingungen hinzuzufügen. Wenn diese Voraussetzungen nicht erfüllt sind, schlägt die Leseanforderung fehl.

Sie können bedingte Lesevorgänge für GETHEAD, oder COPY Anfragen verwenden, um ein Objekt nur auf der Grundlage seiner Metadaten zurückzugeben.

Wenn Sie ein Objekt ablehnen, erstellt Amazon S3 systemgesteuerte Metadaten, die nur von S3 geändert werden können. Entity Tags (ETags) und Last-Modified sind Beispiele für systemgesteuerte Metadaten. Bei einem Objekt ETag handelt es sich um eine Zeichenfolge, die eine bestimmte Version eines Objekts darstellt. Bei dem Datum der letzten Änderung handelt es sich um Metadaten, die das Erstellungsdatum oder das Datum der letzten Änderung eines Objekts angeben, je nachdem, welches Datum das letzte ist.

Bei bedingten Lesevorgängen können Sie ein Objekt auf der Grundlage des Datums der letzten Änderung ETag oder des Datums der letzten Änderung zurückgeben. Sie können in Ihrer Anfrage einen ETag Wert angeben und das Objekt nur zurückgeben, wenn der ETag Wert übereinstimmt. Dadurch kann sichergestellt werden, dass Sie nur eine bestimmte Version eines Objekts zurückgeben oder kopieren. Sie können in Ihrer Leseanforderung den Wert Letzte Änderung angeben und ein Objekt nur zurückgeben, wenn dieses Objekt seit einem von Ihnen angegebenen Datum geändert wurde.

Unterstützt APIs

Die folgende APIs S3-Unterstützung mit bedingten Lesevorgängen:

- [GetObject](#)
- [HeadObject](#)
- [CopyObject](#)

Sie können die folgenden Header verwenden, um ein Objekt zurückzugeben, das vom Entity-Tag (ETag) oder dem Datum der letzten Änderung abhängig ist. Weitere Hinweise zu Objektmetadaten wie ETags Last-Modified finden Sie unter [the section called "Systemdefinierte Objektmetadaten"](#)

[GetObject](#)

- **If-Match**— Gibt das Objekt nur zurück, wenn es ETag mit dem angegebenen übereinstimmt.
- **If-Modified-Since**— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt geändert wurde.
- **If-None-Match**— Gibt das Objekt nur zurück, wenn ETag es nicht mit dem angegebenen übereinstimmt.
- **If-Unmodified-Since**— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt nicht geändert wurde.

Weitere Hinweise zu diesen Headern, den zurückgegebenen Fehlern und der Reihenfolge, in der S3 mehrere bedingte Header in einer einzigen Anfrage verarbeitet, finden Sie unter [GetObject](#) in der Amazon Simple Storage Service API Reference.

[HeadObject](#)

- **If-Match**— Gibt das Objekt nur zurück, wenn es ETag mit dem angegebenen übereinstimmt.
- **If-Modified-Since**— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt geändert wurde.
- **If-None-Match**— Gibt das Objekt nur zurück, wenn ETag es nicht mit dem angegebenen übereinstimmt.
- **If-Unmodified-Since**— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt nicht geändert wurde.

Weitere Hinweise zu diesen Headern, den zurückgegebenen Fehlern und der Reihenfolge, in der S3 mehrere bedingte Header in einer einzigen Anfrage verarbeitet, finden Sie unter [HeadObject](#) in der Amazon Simple Storage Service API Reference.

[CopyObject](#)

- `x-amz-copy-source-if-match`— Kopiert das Quellobjekt nur, wenn es ETag mit dem angegebenen übereinstimmt.
- `x-amz-copy-source-if-modified-since`— Kopiert das Quellobjekt nur, wenn es seit dem angegebenen Zeitpunkt geändert wurde.
- `x-amz-copy-source-if-none-match`— Kopiert das Quellobjekt nur, wenn ETag es nicht mit dem angegebenen übereinstimmt.
- `x-amz-copy-source-if-unmodified-since`— Kopiert das Quellobjekt nur, wenn es seit dem angegebenen Zeitpunkt nicht geändert wurde.

Weitere Hinweise zu diesen Headern, den zurückgegebenen Fehlern und der Reihenfolge, in der S3 mehrere bedingte Header in einer einzigen Anfrage verarbeitet, finden Sie unter [CopyObject](#) in der Amazon Simple Storage Service API Reference.

Wie verhindert man Objektüberschreibungen durch bedingte Schreibvorgänge

Bei bedingten Schreibvorgängen können Sie Ihrer Schreibanforderung einen zusätzlichen Header hinzufügen, um Vorbedingungen für Ihren S3-Vorgang festzulegen. Auf diese Weise können Sie das Überschreiben vorhandener Daten verhindern, indem überprüft wird, ob sich in Ihrem Bucket noch kein Objekt mit demselben Schlüsselnamen befindet. Bedingte Schreibvorgänge funktionieren für API Anfragen in Allzweck-Buckets und Verzeichnis-Buckets.

Wenn Sie ein Objekt auf Amazon S3 hochladen, geben Sie den Schlüsselnamen an. Der Schlüsselname ist eine eindeutige Kennung eines Objekts in einem Bucket, bei der Groß- und Kleinschreibung berücksichtigt wird. Wenn Sie ein Objekt mit einem identischen Schlüsselnamen in einen Bucket hochladen, für den keine Version verwendet wurde oder dessen Version gesperrt wurde, wird das Objekt überschrieben. In einem versionierten Bucket wird das zuletzt hochgeladene Objekt zur aktuellen Version des Objekts.

Bei bedingten Schreibvorgängen wird während des WRITE Vorgangs geprüft, ob ein Objekt vorhanden ist. Wenn im Bucket ein identischer Schlüsselname gefunden wird, schlägt der WRITE

Vorgang fehl. Bei bedingten Schreibvorgängen können mehrere Clients in denselben Bucket schreiben, ohne dass die Möglichkeit besteht, bestehende Objekte zu überschreiben.

Um bedingte Schreibvorgänge ausführen zu können, benötigen Sie die `s3:PutObject` entsprechende Genehmigung. Diese Berechtigung ermöglicht es dem Aufrufer, zu überprüfen, ob Objekte im Bucket vorhanden sind. Sie können bedingte Schreibvorgänge mit vorsigniertem Befehl URLs verwenden AWS SDKs.

Note

Um bedingte Schreibvorgänge zu verwenden, müssen Sie die Anfragen über HTTPS (TLS) stellen oder verwenden AWS Signaturversion 4, um die Anfrage zu signieren.

Unterstützt APIs

Die folgende APIs S3-Unterstützung mit bedingten Schreibvorgängen:

- [PutObject](#)
- [CompleteMultipartUpload](#)

Sie können die folgenden Header verwenden, um ein Objekt zu schreiben, das vom Objektschlüsselnamen abhängig ist. Hinweise zu Objektschlüsselnamen finden Sie unter [the section called "Erstellen von Objektschlüsseln"](#).

[PutObject](#)

- `If-None-Match`— Lädt das Objekt nur hoch, wenn im angegebenen Bucket noch kein Objekt mit demselben Schlüsselnamen existiert. Sie müssen den Wert `*` (Sternchen) mit diesem Parameter verwenden.

Verwendung der AWS CLI

Der folgende `put-object` Beispielbefehl zeigt, wie Sie AWS CLI um ein Objekt mit einem Conditional Write-Header mithilfe des `if-none-match` Parameters hochzuladen.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key dir-1/my_images.tar.bz2 --body my_images.tar.bz2 --if-none-match "*" 
```

Weitere Informationen finden Sie unter [put-object](#) in der AWS CLI Befehlsreferenz.

Für Informationen über AWS CLI, siehe [Was ist der AWS Command Line Interface?](#) in der AWS Command Line Interface Benutzerleitfaden.

Weitere Informationen zu diesen Headern finden Sie unter [PutObject](#) in der Amazon Simple Storage Service API Reference.

[CompleteMultipartUpload](#)

- **If-None-Match**— Schließt den Upload nur ab, wenn im angegebenen Bucket noch kein Objekt mit demselben Schlüsselnamen vorhanden ist. Sie müssen den Wert * (Sternchen) für diesen Parameter verwenden.

Verwendung der AWS CLI

Der folgende `complete-multipart-upload` Beispielfehl zeigt, wie Sie AWS CLI um einen mehrteiligen Upload mit einem Conditional Write-Header mithilfe des `if-none-match` Parameters abzuschließen.

```
aws s3api complete-multipart-upload --multipart-upload file://mpustruct --bucket amzn-s3-demo-bucket --key dir-1/my_images.tar.bz2 --upload-id uploadID --if-none-match "*" 
```

Weitere Informationen finden Sie unter [complete-multipart-upload](#) in der AWS CLI Befehlsreferenz.

Für Informationen über AWS CLI, siehe [Was ist der AWS Command Line Interface?](#) in der AWS Command Line Interface Benutzerleitfaden.

Weitere Informationen zu diesen Headern finden Sie unter [CompleteMultipartUpload](#) in der Amazon Simple Storage Service API Reference.

Bedingtes Schreibverhalten

Bedingte Schreibvorgänge werden anhand vorhandener Objekte in einem Bucket ausgewertet. Wenn im Bucket kein Objekt mit demselben Schlüsselnamen vorhanden ist, ist der Schreibvorgang erfolgreich, was zu einer `200` Antwort führt. Wenn ein Objekt vorhanden ist, schlägt der Schreibvorgang fehl, was zu einer `412 Precondition Failed` Antwort führt. Bei Buckets mit aktivierter Versionierung prüft S3 im Rahmen der bedingten Bewertung, ob eine aktuelle

Objektversion mit demselben Namen vorhanden ist. Wenn es keine aktuelle Objektversion mit demselben Namen gibt oder wenn es sich bei der aktuellen Objektversion um eine Löschmarke handelt, ist der Schreibvorgang erfolgreich. Andernfalls führt dies zu einem fehlgeschlagenen Schreibvorgang mit einer `412 Precondition Failed` Antwort.

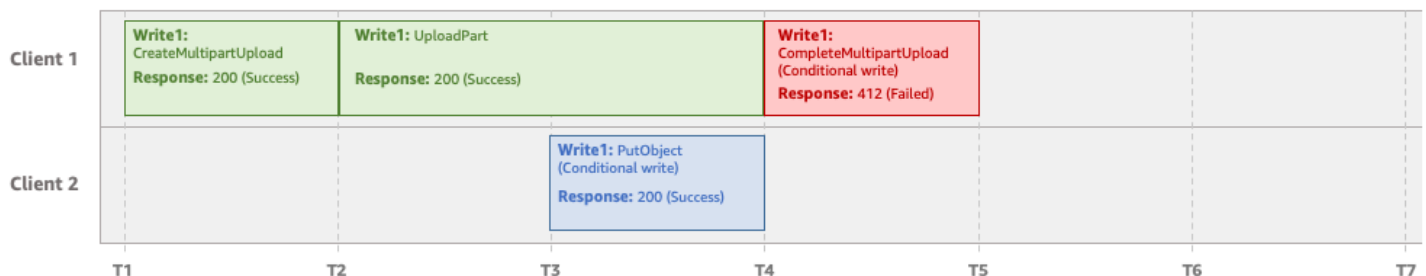
Wenn mehrere bedingte Schreibvorgänge für denselben Objektnamen ausgeführt werden, ist der erste abgeschlossene Schreibvorgang erfolgreich. Amazon S3 schlägt dann nachfolgende Schreibvorgänge mit einer `412 Precondition Failed` Antwort fehl.

Sie können auch bei gleichzeitigen Anfragen eine `409 Conflict` Antwort erhalten, wenn eine Löschanforderung für ein Objekt erfolgreich ist, bevor ein bedingter Schreibvorgang für dieses Objekt abgeschlossen werden kann. Das liegt daran, dass die Löschanforderung Vorrang vor dem zuvor initiierten bedingten Schreibvorgang hat. Wenn Sie bedingte Schreibvorgänge mit `writePutObject` verwenden, werden Uploads möglicherweise erneut versucht, nachdem ein Fehler aufgetreten ist. `409` Bei Verwendung `CompleteMultipartUpload` muss der mehrteilige Upload erneut initiiert werden, um das Objekt nach Erhalt eines Fehlers erneut hochzuladen. `CreateMultipartUpload 409`

Stellen Sie sich die folgenden Szenarien vor, in denen zwei Clients Operationen auf demselben Bucket ausführen.

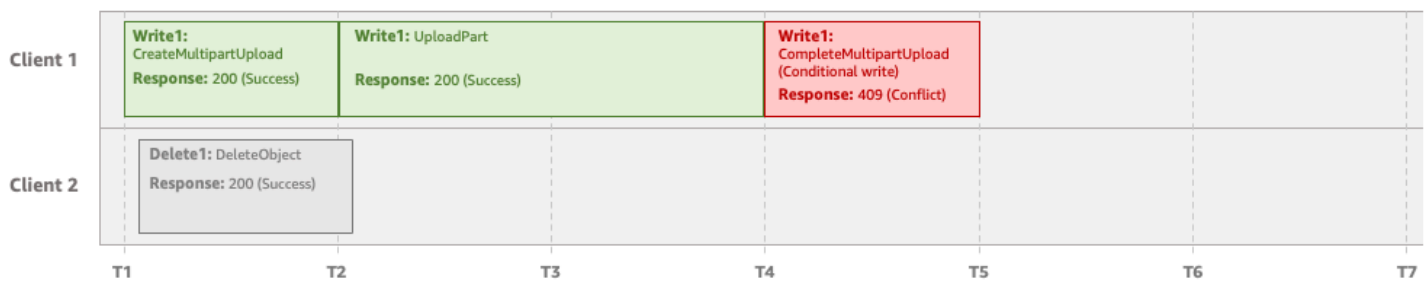
4.1.2 Vorbedingung: Fehlgeschlagene Antwort

Bei bedingten Schreibvorgängen werden Anfragen für mehrteilige Uploads nicht berücksichtigt, da es sich bei diesen noch nicht um vollständig geschriebene Objekte handelt. Stellen Sie sich das folgende Beispiel vor, in dem Client 1 ein Objekt mithilfe eines mehrteiligen Uploads hochlädt. Während des mehrteiligen Uploads ist Client 2 in der Lage, dasselbe Objekt mit dem bedingten Schreibvorgang erfolgreich zu schreiben. Wenn Client 1 anschließend versucht, den mehrteiligen Upload mithilfe eines bedingten Schreibvorgangs abzuschließen, schlägt der Upload fehl, da das Objekt bereits existiert.



409 Reaktion auf Konflikte

Wenn eine Löschanforderung erfolgreich ist, bevor eine bedingte Schreibanforderung abgeschlossen werden kann, gibt Amazon S3 eine `409 Conflict` Antwort für den Schreibvorgang zurück. Das liegt daran, dass die zuvor initiierte Löschanforderung Vorrang vor dem bedingten Schreibvorgang hat. Betrachten Sie das folgende Beispiel, in dem eine Datei in einem Bucket `puppy.txt` existiert. Client 1 startet einen mehrteiligen Upload für eine andere Datei, die ebenfalls benannt ist, `puppy.txt` mit der Absicht, den mehrteiligen Upload mit einem bedingten Schreibvorgang abzuschließen. Während des Uploads löscht Client 2 `puppy.txt` aus dem Bucket. Wenn Client 1 versucht, seine eigene `puppy.txt` Datei mit einem bedingten Schreibvorgang hochzuladen, schlägt dies fehl und es wird eine `409 Conflict` Antwort angezeigt. `CompleteMultipartUpload` In solchen Fällen müssen Sie einen neuen mehrteiligen Upload initiieren.



Note

Wir empfehlen, eine Lebenszyklusregel zu konfigurieren, durch die unvollständige mehrteilige Uploads nach einer bestimmten Anzahl von Tagen mit der Aktion `AbortIncompleteMultipartUpload` gelöscht werden, um Ihre Speicherkosten gering zu halten. Weitere Informationen zum Erstellen einer Lebenszyklusregel zum Löschen unvollständiger mehrteiliger Uploads finden Sie unter [Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads](#).


Objekte kopieren, verschieben und umbenennen

Der `CopyObject` Vorgang erstellt eine Kopie eines Objekts, das bereits in Amazon S3 gespeichert ist.

Sie können in einem einzigen atomaren Vorgang eine Kopie eines Objekts mit einer Größe von bis zu 5 GB erstellen. Um ein Objekt zu kopieren, das größer als 5 GB ist, müssen Sie jedoch einen mehrteiligen Upload verwenden. Weitere Informationen finden Sie unter [the section called "Kopieren eines Objekts"](#).

Mit der CopyObject-Operation können Sie:

- Erstellen Sie zusätzliche Kopien von Objekten.
- Benennen Sie Objekte um, indem Sie sie kopieren und die Originale löschen.
- Kopieren oder verschieben Sie Objekte von einem Bucket in einen anderen, auch zwischen AWS-Regionen (z. B. von us-west-1 bis eu-west-2). Wenn Sie ein Objekt verschieben, kopiert Amazon S3 das Objekt an das angegebene Ziel und löscht dann das Quellobjekt.

 Note

Objekte kopieren oder verschieben zwischen AWS-Regionen verursacht Bandbreitengebühren. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

- Ändern Sie die Metadaten des Objekts. Jedes Amazon-S3-Objekt hat Metadaten. Bei diesen Metadaten handelt es sich um eine Reihe von Name-Wert-Paaren. Sie können Objektmetadaten festlegen, wenn Sie ein Objekt hochladen. Nachdem Sie das Objekt hochgeladen haben, können Sie die Objektmetadaten nicht mehr ändern. Die einzige Methode, wie Sie Objekt-Metadaten ändern können, ist es, eine Kopie des Objekts anzulegen und die Metadaten festzulegen. Stellen Sie dazu beim Kopiervorgang dasselbe Objekt als Quelle und Ziel ein.

Einige Objektmetadaten sind Systemmetadaten und andere sind benutzerdefiniert. Sie können einige der Systemmetadaten steuern. Sie können beispielsweise die Speicherklasse und die Art der serverseitigen Verschlüsselung steuern, die für das Objekt verwendet werden sollen. Wenn Sie ein Objekt kopieren, werden auch die Systemmetadaten und benutzerdefinierte Metadaten kopiert. Amazon S3 setzt die vom System gesteuerten Metadaten zurück. Wenn Sie beispielsweise ein Objekt kopieren, setzt Amazon S3 das Erstellungsdatum des kopierten Objekts zurück. Sie müssen in Ihrer Kopieranforderung keinen dieser systemgesteuerten Metadatenwerte festlegen.

Wenn Sie ein Objekt kopieren, wollen Sie möglicherweise einige der Metadatenwerte aktualisieren. Wenn Ihr Quellobjekt beispielsweise für die Verwendung des S3-StandardSpeichers konfiguriert ist, könnten Sie entscheiden, für die Objektkopie S3 Intelligent Tiering zu verwenden. Außerdem könnten Sie entscheiden, einige der benutzerdefinierten Metadatenwerte für das Quellobjekt zu aktualisieren. Wenn Sie entschieden haben, beim Kopieren vom Benutzer des Objekts konfigurierbare Metadaten zu aktualisieren (System oder benutzerdefiniert), müssen Sie explizit alle vom Benutzer konfigurierbaren Metadaten angeben, die im Quellobjekt Ihrer Anforderung vorhanden sind, selbst wenn Sie nur einen der Metadatenwerte ändern.

Weitere Informationen zu Objekt-Metadaten erhalten Sie unter [Arbeiten mit Objekt-Metadaten](#).

Archivierte und wiederhergestellte Objekte werden kopiert

Wenn das Quellobjekt in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert ist, müssen Sie zuerst eine temporäre Kopie wiederherstellen, bevor Sie das Objekt in einen anderen Bucket kopieren können. Weitere Informationen über das Archivieren von Objekten finden Sie unter [Übergang in die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive \(Objektarchivierung\)](#).

Der Kopiervorgang in der Amazon S3 S3-Konsole wird für wiederhergestellte Objekte in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive nicht unterstützt. Um diese wiederhergestellten Objekte zu kopieren, verwenden Sie AWS Command Line Interface (AWS CLI), das AWS SDKs, oder der Amazon S3 RESTAPI.

Verschlüsselte Objekte werden kopiert

Amazon S3 verschlüsselt alle neuen Objekte, die in einen S3-Bucket kopiert werden, automatisch. Wenn Sie in Ihrer Kopieranforderung keine Verschlüsselungsinformationen angeben, wird die Verschlüsselungseinstellung des Zielobjekts auf die Standardverschlüsselungskonfiguration des Ziel-Buckets festgelegt. Standardmäßig haben alle Buckets eine grundlegende Verschlüsselungskonfiguration, die serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwendet. Wenn der Ziel-Bucket über eine Standard-Verschlüsselungskonfiguration verfügt, die serverseitige Verschlüsselung mit einem verwendet AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder einen vom Kunden bereitgestellten Verschlüsselungsschlüssel (SSE-C), Amazon S3 verwendet den entsprechenden KMS Schlüssel oder einen vom Kunden bereitgestellten Schlüssel, um die Zielobjektkopie zu verschlüsseln.

Wenn Sie beim Kopieren eines Objekts eine andere Art der Verschlüsselungseinstellung für das Zielobjekt verwenden möchten, können Sie verlangen, dass Amazon S3 das Zielobjekt mit einem KMS Schlüssel, einem von Amazon S3 verwalteten Schlüssel oder einem vom Kunden bereitgestellten Schlüssel verschlüsselt. Wenn sich die Verschlüsselungseinstellung in Ihrer Anforderung von der Standardverschlüsselungskonfiguration des Ziel-Buckets unterscheidet, hat die Verschlüsselungseinstellung in Ihrer Anfrage Vorrang. Wenn das Quellobjekt für die Kopie mit SSE -C verschlüsselt ist, müssen Sie in Ihrer Anfrage die erforderlichen Verschlüsselungsinformationen angeben, damit Amazon S3 das Objekt zum Kopieren entschlüsseln kann. Weitere Informationen finden Sie unter [Datenschutz durch Verschlüsselung](#).

Verwendung von Prüfsummen beim Kopieren von Objekten

Beim Kopieren von Objekten können Sie einen anderen Prüfsummenalgorithmus für das Objekt verwenden. Unabhängig davon, ob Sie denselben oder einen neuen Algorithmus verwenden, berechnet Amazon S3 einen neuen Prüfsummenwert, nachdem das Objekt kopiert wurde. Amazon S3 kopiert den Wert der Prüfsumme nicht direkt. Der Prüfsummenwert von Objekten, die mithilfe von mehrteiligen Uploads geladen wurden, kann sich ändern. Weitere Informationen zum Berechnen der Prüfsumme finden Sie unter [Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads](#).

Kopieren mehrerer Objekte in einer einzigen Anfrage

Um mehr als ein Amazon S3 S3-Objekt mit einer einzigen Anfrage zu kopieren, können Sie auch S3 Batch Operations verwenden. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, für die Vorgänge ausgeführt werden sollen. S3 Batch Operations ruft die entsprechende API Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Die Funktion „S3-Batchoperationen“ verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. Sie profitieren von einer vollständig verwalteten, prüfbar und serverlosen Umgebung. Sie können S3 Batch Operations über die Amazon S3 S3-Konsole verwenden. AWS CLI, AWS SDKs, oder RESTAPI. Weitere Informationen finden Sie unter [the section called “Grundlagen von BatchVorgänge”](#).

Objekte in Verzeichnis-Buckets kopieren

Hinweise zum Kopieren eines Objekts in einen Verzeichnis-Bucket finden Sie unter [Hochladen eines Objekts zu einem Verzeichnis-Bucket](#). Informationen zur Verwendung der Amazon S3 Express One Zone-Speicherklasse mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Ein Objekt kopieren

Verwenden Sie die folgenden Methoden, um ein Objekt zu kopieren.

Verwenden der S3-Konsole

Note

- Wenn Sie ein Objekt mithilfe der Amazon S3 S3-Konsole kopieren möchten, benötigen Sie die `s3:ListAllMyBuckets` entsprechende Genehmigung. Die Konsole benötigt diese

Berechtigung, um den Kopiervorgang zu validieren. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unter [the section called “Beispiele für identitätsbasierte Richtlinien”](#).

Wenn Sie ein Objekt kopieren, das über benutzerdefinierte Tags verfügt, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Berechtigung. Wenn Sie ein Objekt kopieren, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` -Berechtigung.

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, wird das Objekt ohne die benutzerdefinierten Tags kopiert, und Sie erhalten eine Fehlermeldung.

- Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mit der S3-Konsole kopiert werden. Um mit SSE -C verschlüsselte Objekte zu kopieren, verwenden Sie den AWS CLI, AWS SDK, oder der Amazon S3 RESTAPI.
- Regionsübergreifendes Kopieren von Objekten, die mit SSE — verschlüsselt sind, KMS wird von der Amazon S3 S3-Konsole nicht unterstützt. Um Objekte, die mit SSE — verschlüsselt sind, KMS zwischen Regionen zu kopieren, verwenden Sie den AWS CLI, AWS SDK, oder der Amazon S3 RESTAPI.

Ein Objekt kopieren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zum Amazon-S3-Bucket oder -Ordner, der die Objekte enthält, die Sie kopieren möchten.
3. Markieren Sie das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.
4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Kopieren aus.
5. Wählen Sie den Zieltyp und das Zielkonto aus. Um den Zielpfad anzugeben, wählen Sie Browse S3 (S3 durchsuchen), navigieren Sie zum Ziel und markieren Sie das Kontrollkästchen links von dem Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

Geben Sie alternativ den Zielpfad ein.

6. Wenn Sie die Bucket-Versionierung nicht aktiviert haben, werden Sie möglicherweise aufgefordert, zu bestätigen, dass vorhandene Objekte mit demselben Namen überschrieben werden. Wenn dies in Ordnung ist, markieren Sie das Kontrollkästchen und fahren Sie fort. Wenn Sie alle Versionen von Objekten in diesem Bucket behalten möchten, wählen Sie Enable Bucket Versioning (Bucket-Versionierung aktivieren). Sie können auch die Eigenschaften der Standard-Verschlüsselung und der S3-Objektsperre aktualisieren.
7. Wählen Sie unter Additional checksums (Weitere Prüfsummen) aus, ob Sie die Objekte mit der vorhandenen Prüfsummenfunktion kopieren oder die vorhandene Prüfsummenfunktion durch eine neue ersetzen möchten. Beim Hochladen der Objekte hatten Sie die Möglichkeit, den Prüfsummenalgorithmus anzugeben, der zur Überprüfung der Datenintegrität verwendet wurde. Beim Kopieren des Objekts haben Sie die Möglichkeit, eine neue Funktion auszuwählen. Wenn Sie ursprünglich keine weitere Prüfsumme angegeben haben, können Sie diesen Abschnitt der Kopieroptionen verwenden, um eine Summe hinzuzufügen.

Note

Selbst wenn Sie dieselbe Prüfsummenfunktion verwenden, kann sich Ihr Prüfsummenwert ändern, wenn Sie das Objekt kopieren und dessen Größe 16 MB überschreitet. Der Prüfsummenwert kann sich aufgrund der Methode ändern, wie Prüfsummen für mehrteilige Uploads berechnet werden. Weitere Informationen dazu, wie sich die Prüfsumme beim Kopieren des Objekts ändern kann, finden Sie unter [Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads](#).

Um die Prüfsummenfunktion zu ändern, wählen Sie Replace with a new checksum function (Durch eine neue Prüfsummenfunktion ersetzen) aus. Wählen Sie die neue Prüfsummenfunktion aus dem Feld aus. Wenn das Objekt kopiert wird, wird die neue Prüfsumme mit dem angegebenen Algorithmus berechnet und gespeichert.

8. Wählen Sie unten rechts Copy (Kopieren) aus. Amazon S3 kopiert Ihre Objekte in den Zielordner.

Verwendung der AWS SDKs

Die Beispiele in diesem Abschnitt zeigen, wie Objekte mit bis zu 5 GB in einer einzigen Operation kopiert werden können. Um Objekte zu kopieren, die größer als 5 GB sind, müssen Sie einen mehrteiligen Upload verwenden. Weitere Informationen finden Sie unter [Kopieren eines Objekts mit Multipart-Upload](#).

Java

Example

Das folgende Beispiel kopiert ein Objekt in Amazon S3 mit dem AWS SDK for Java. Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

import java.io.IOException;

public class CopyObjectSingleOperation {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String sourceKey = "**** Source object key *** ";
        String destinationKey = "**** Destination object key ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjRequest = new CopyObjectRequest(bucketName,
sourceKey, bucketName, destinationKey);
            s3Client.copyObject(copyObjRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
```

```
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Im folgenden C#-Beispiel wird das High-Level verwendete AWS SDK für .NET um Objekte mit einer Größe von bis zu 5 GB in einem einzigen Vorgang zu kopieren. Für Objekte mit mehr als 5 GB müssen Sie das in [Kopieren eines Objekts mit Multipart-Upload](#) beschriebene Kopierbeispiel für mehrteilige Uploads verwenden.

In diesem Beispiel wird eine Kopie eines Objekts bis zu einer Größe von 5 GB erstellt. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
source object ***";
        private const string destinationBucket = "*** provide the name of the bucket
to copy the object to ***";
        private const string objectKey = "*** provide the name of object to copy
***";
        private const string destObjectKey = "*** provide the destination object key
name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
```

```
    {
        s3Client = new AmazonS3Client(bucketRegion);
        Console.WriteLine("Copying an object");
        CopyingObjectAsync().Wait();
    }

private static async Task CopyingObjectAsync()
{
    try
    {
        CopyObjectRequest request = new CopyObjectRequest
        {
            SourceBucket = sourceBucket,
            SourceKey = objectKey,
            DestinationBucket = destinationBucket,
            DestinationKey = destObjectKey
        };
        CopyObjectResponse response = await
s3Client.CopyObjectAsync(request);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}
}
```

PHP

Dieses Thema führt Sie durch die Verwendung von Klassen aus Version 3 von AWS SDK for PHP um ein einzelnes Objekt und mehrere Objekte innerhalb von Amazon S3, von einem Bucket in einen anderen oder innerhalb desselben Buckets zu kopieren.

Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Das folgende PHP Beispiel veranschaulicht die Verwendung der `copyObject()` Methode zum Kopieren eines einzelnen Objekts innerhalb von Amazon S3. Es zeigt auch, wie Sie mehrere Kopien eines Objekts erstellen können, indem Sie eine Reihe von Aufrufen `CopyObject` mit der `getCommand()` Methode verwenden.

Objekte kopieren

- 1 Erstellen Sie mit dem Klassenkonstruktor `Aws\S3\S3Client` eine Instance eines Amazon-S3-Clients.
- 2 Um mehrere Kopien eines Objekts zu erstellen, führen Sie eine Reihe von Aufrufen an den Amazon S3 S3-Client aus. `getCommand()` Methode, die von der geerbt wird `Aws\CommandInterface` Klasse. Sie stellen den `CopyObject` -Befehl als erstes Argument, sowie ein Array mit dem Quell-Bucket, dem Quellschlüsselnamen, dem Ziel-Bucket und dem Zielschlüsselnamen als zweites Argument bereit.

```
require 'vendor/autoload.php';

use Aws\CommandPool;
use Aws\Exception\AwsException;
use Aws\ResultInterface;
use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';
$targetBucket = '*** Your Target Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Copy an object.
$s3->copyObject([
    'Bucket' => $targetBucket,
    'Key' => "$sourceKeyname-copy",
    'CopySource' => "$sourceBucket/$sourceKeyname",
]);

// Perform a batch of CopyObject operations.
$batch = array();
```

```

for ($i = 1; $i <= 3; $i++) {
    $batch[] = $s3->getCommand('CopyObject', [
        'Bucket' => $targetBucket,
        'Key' => "{targetKeyname}-$i",
        'CopySource' => "$sourceBucket/$sourceKeyname",
    ]);
}
try {
    $results = CommandPool::batch($s3, $batch);
    foreach ($results as $result) {
        if ($result instanceof ResultInterface) {
            // Result handling here
        }
        if ($result instanceof AwsException) {
            // AwsException handling here
        }
    }
} catch (Exception $e) {
    // General error handling here
}

```

Python

```

class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
        Boto3
                           that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

```

```

def copy(self, dest_object):
    """
    Copies the object to another bucket.

    :param dest_object: The destination object initialized with a bucket and
    key.

```

```
        """
        This is a Boto3 Object resource.
        """
        try:
            dest_object.copy_from(
                CopySource={"Bucket": self.object.bucket_name, "Key":
self.object.key}
            )
            dest_object.wait_until_exists()
            logger.info(
                "Copied object from %s:%s to %s:%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
        except ClientError:
            logger.exception(
                "Couldn't copy object from %s/%s to %s/%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
            raise
```

Ruby

Die folgenden Aufgaben führen Sie durch die Verwendung des Ruby Klassen zum Kopieren eines Objekts in Amazon S3 von einem Bucket in einen anderen oder innerhalb desselben Buckets.

Objekte kopieren

- 1 Verwenden Sie das modularisierte Amazon S3 S3-Gem für Version 3 von AWS SDK for Ruby, benötigen und `aws-sdk-s3` stellen Sie Ihre AWS Anmeldeinformationen. Weitere Informationen zur Bereitstellung Ihrer Anmeldeinformationen finden Sie in [Anfragen stellen mit AWS-Konto oder IAM Benutzeranmeldedaten](#).
- 2 Geben Sie die Anforderungsinformationen an, z. B. den Quell-Bucket-Namen, den Quell-Bucket-Namen, den Ziel-Bucket-Namen und den Zielschlüssel.

Folgendes Ruby Das Codebeispiel veranschaulicht die vorherigen Aufgaben, indem die `#copy_object` Methode verwendet wird, um ein Objekt von einem Bucket in einen anderen zu kopieren.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
  # used as the source object for
  #
  #           copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket and rename it with the
  # target key.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
  # object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
  # nil.
  def copy_object(target_bucket, target_object_key)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
    #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
end
```

```

target_bucket = Aws::S3::Bucket.new(target_bucket_name)
target_object = wrapper.copy_object(target_bucket, target_key)
return unless target_object

puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__

```

Unter Verwendung des REST API

Dieses Beispiel beschreibt, wie ein Objekt mithilfe von Amazon S3 kopiert wird RESTAPI. Weitere Informationen über das finden REST API Sie unter [CopyObject](#).

Dieses Beispiel kopiert das `flotsam`-Objekt aus dem `amzn-s3-demo-bucket1`-Bucket in das `jetsam`-Objekt des `amzn-s3-demo-bucket2`-Buckets, wobei seine Metadaten beibehalten werden.

```

PUT /jetsam HTTP/1.1
Host: amzn-s3-demo-bucket2.s3.amazonaws.com
x-amz-copy-source: /amzn-s3-demo-bucket1/flotsam
Authorization: AWS AKIAIOSFODNN7EXAMPLE:ENoSbxYByFA0UGLZUqJN5EUnLDg=
Date: Wed, 20 Feb 2008 22:12:21 +0000

```

Die Signatur wurde aus den folgenden Informationen generiert.

```

PUT\r\n
\r\n
\r\n
Wed, 20 Feb 2008 22:12:21 +0000\r\n

x-amz-copy-source:/amzn-s3-demo-bucket1/flotsam\r\n
/amzn-s3-demo-bucket2/jetsam

```

Amazon S3 gibt die folgende Antwort zurück, in ETag der das Objekt und das Datum der letzten Änderung angegeben sind.

```

HTTP/1.1 200 OK
x-amz-id-2: Vyaxt7qEbvz34BnSu5hctyyNSlHTYZFMWK4Ftz0+iX8JQNyaLdTshL0Kxatba0Zt
x-amz-request-id: 6B13C3C5B34AF333
Date: Wed, 20 Feb 2008 22:13:01 +0000

```

```
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>

<CopyObjectResult>
  <LastModified>2008-02-20T22:13:01</LastModified>
  <ETag>"7e9c608af58950deeb370c98608ed097"</ETag>
</CopyObjectResult>
```

Verwendung der AWS CLI

Sie können auch das verwenden AWS Command Line Interface (AWS CLI), um ein S3-Objekt zu kopieren. Weitere Informationen finden Sie unter [copy-object](#) in der AWS CLI Befehlsreferenz.

Für Informationen über AWS CLI, siehe [Was ist der AWS Command Line Interface?](#) in der AWS Command Line Interface Benutzerleitfaden.

So verschieben Sie ein Objekt

Verwenden Sie die folgenden Methoden, um ein Objekt zu verschieben.

Verwenden der S3-Konsole

Note

- Wenn Sie ein Objekt mit benutzerdefinierten Tags verschieben, benötigen Sie die entsprechende `s3:GetObjectTagging` Genehmigung. Wenn Sie ein Objekt verschieben, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` -Berechtigung.

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, wird das Objekt ohne die benutzerdefinierten Tags verschoben, und Sie erhalten eine Fehlermeldung.

- Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mithilfe der Amazon S3 S3-Konsole verschoben werden. Um mit SSE -C verschlüsselte Objekte zu verschieben, verwenden Sie den AWS CLI, AWS SDKs, oder der Amazon S3 RESTAPI.

- Warten Sie beim Verschieben von Ordnern, bis der Verschiebevorgang abgeschlossen ist, bevor Sie weitere Änderungen an den Ordnern vornehmen.
- Sie können S3-Zugriffspunkt-Aliase nicht als Quelle oder Ziel für Move-Operationen in der Amazon S3 S3-Konsole verwenden.

So verschieben Sie ein Objekt

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zum Amazon-S3-Bucket oder -Ordner, der die Objekte enthält, die Sie verschieben möchten.
3. Markieren Sie das Kontrollkästchen links neben den Namen der Objekte, die Sie verschieben möchten.
4. Wählen Sie im Menü Aktionen die Option Verschieben aus.
5. Um den Zielpfad anzugeben, wählen Sie Browse S3 (S3 durchsuchen), navigieren Sie zum Ziel und markieren Sie das Kontrollkästchen links von dem Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

Geben Sie alternativ den Zielpfad ein.

6. Wenn Sie die Bucket-Versionierung nicht aktiviert haben, werden Sie möglicherweise aufgefordert, zu bestätigen, dass vorhandene Objekte mit demselben Namen überschrieben werden. Wenn dies in Ordnung ist, markieren Sie das Kontrollkästchen und fahren Sie fort. Wenn Sie alle Versionen von Objekten in diesem Bucket behalten möchten, wählen Sie Enable Bucket Versioning (Bucket-Versionierung aktivieren). Sie können auch die Eigenschaften der Standard-Verschlüsselung und der Objektsperre aktualisieren.
7. Wählen Sie unten rechts Move (Verschieben) aus. Amazon S3 verschiebt Ihre Objekte in den Zielordner.

Note

- Diese Aktion erstellt eine Kopie aller angegebenen Objekte mit aktualisierten Einstellungen, aktualisiert das Datum der letzten Änderung am angegebenen Speicherort und fügt dem ursprünglichen Objekt eine Löschmarkierung hinzu.

- Diese Aktion aktualisiert Metadaten für Bucket-Versioning, Verschlüsselung, Objektsperre-Funktionen und archivierte Objekte.

Verwendung der AWS CLI

Sie können auch die AWS Command Line Interface (AWS CLI), um ein S3-Objekt zu verschieben. Weitere Informationen finden Sie unter [mv](#) in der AWS CLI Befehlsreferenz.

Für Informationen über AWS CLI, siehe [Was ist der AWS Command Line Interface?](#) in der AWS Command Line Interface Benutzerleitfaden.

Um ein Objekt umzubenennen

Gehen Sie wie folgt vor, um ein Objekt umzubenennen.

Note

- Beim Umbenennen eines Objekts wird eine Kopie des Objekts mit einem neuen Datum der letzten Änderung erstellt. Anschließend wird dem ursprünglichen Objekt eine Löschmarke hinzugefügt.
- Bucket-Einstellungen für die Standardverschlüsselung werden automatisch auf jedes angegebene unverschlüsselte Objekt angewendet.
- Sie können die Amazon S3 S3-Konsole nicht verwenden, um Objekte mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) umzubenennen. Um mit SSE - C verschlüsselte Objekte umzubenennen, verwenden Sie AWS CLI, AWS SDKs, oder Amazon S3, REST API um diese Objekte mit neuen Namen zu kopieren.
- Wenn dieser Bucket die vom Bucket-Besitzer erzwungene Einstellung für S3 Object Ownership verwendet, werden Objektzugriffskontrolllisten (ACLs) nicht kopiert.
- Wenn Sie ein Objekt umbenennen, das über benutzerdefinierte Tags verfügt, benötigen Sie die entsprechende Genehmigung. `s3:GetObjectTagging` Wenn Sie ein Objekt umbenennen, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3: -Berechtigung. GetObjectTagging`

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, wird das Objekt umbenannt, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

Um ein Objekt umzubenennen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der das Objekt enthält, das Sie umbenennen möchten.
3. Aktivieren Sie das Kontrollkästchen links neben dem Namen des Objekts, das Sie umbenennen möchten.
4. Wählen Sie im Menü Aktionen die Option Objekt umbenennen.
5. Geben Sie im Feld Neuer Objektname den neuen Namen für das Objekt ein.
6. Wählen Sie in der unteren rechten Ecke die Option Änderungen speichern aus. Amazon S3 benennt Ihr Objekt um.

Herunterladen von Objekten

In diesem Abschnitt wird erläutert, wie Sie Objekte aus einem S3-Bucket herunterladen. Mit Amazon S3 können Sie diese Objekte in einem oder mehreren Buckets speichern. Jedes Objekt kann eine Größe von bis zu 5 TB haben. Der Zugriff auf Amazon-S3-Objekte, die nicht archiviert sind, ist in Echtzeit möglich. Archivierte Objekte müssen jedoch wiederhergestellt werden, bevor sie heruntergeladen werden können. Weitere Informationen zum Herunterladen von archivierten Objekten finden Sie unter [the section called “Herunterladen von archivierten Objekten”](#).

Sie können ein einzelnes Objekt mithilfe der Amazon S3 S3-Konsole herunterladen. AWS Command Line Interface (AWS CLI), AWS SDKs oder Amazon S3 RESTAPI. Verwenden Sie die S3-Konsole, um ein Objekt von S3 herunterzuladen, ohne Code schreiben oder Befehle ausführen zu müssen. Weitere Informationen finden Sie unter [the section called “Herunterladen eines Objekts”](#).

Um mehrere Objekte herunterzuladen, verwenden Sie AWS CloudShell, das AWS CLI, oder der AWS SDKs. Weitere Informationen finden Sie unter [the section called “Herunterladen mehrerer Objekte”](#).

Wenn Sie einen Teil eines Objekts herunterladen müssen, verwenden Sie zusätzliche Parameter mit AWS CLI oder REST API um nur die Bytes anzugeben, die Sie herunterladen möchten. Weitere Informationen finden Sie unter [the section called “Herunterladen eines Teils eines Objekts”](#).

Wenn Sie ein Objekt herunterladen müssen, das Sie nicht besitzen, bitten Sie den Objekteigentümer, ein vorsigniertes Objekt zu generierenURL, mit dem Sie das Objekt herunterladen können. Weitere

Informationen finden Sie unter [the section called “Ein Objekt von einem anderen herunterladen AWS-Konto”](#).

Wenn Sie Objekte außerhalb von herunterladen AWS Netzwerk, Datenübertragungsgebühren fallen an. Datenübertragung innerhalb des AWS Das Netzwerk ist innerhalb desselben kostenlos AWS-Region, aber alle GET Anfragen werden Ihnen in Rechnung gestellt. Weitere Informationen zu den Gebühren für Datenübertragungen und Datenabrufe finden Sie unter [Preise für Amazon S3](#).

Themen

- [Herunterladen eines Objekts](#)
- [Herunterladen mehrerer Objekte](#)
- [Herunterladen eines Teils eines Objekts](#)
- [Ein Objekt von einem anderen herunterladen AWS-Konto](#)
- [Herunterladen von archivierten Objekten](#)
- [Objekte auf der Grundlage von Metadaten herunterladen](#)
- [Fehlerbehebung beim Herunterladen von Objekten](#)

Herunterladen eines Objekts

Sie können ein Objekt mithilfe der Amazon S3 S3-Konsole herunterladen. AWS CLI, AWS SDKs, oder RESTAPI.

Verwenden der S3-Konsole

In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole ein Objekt aus einem S3-Bucket herunterladen.

Note

- Sie können jeweils nur ein Objekt herunterladen.
- Wenn Sie über die Amazon-S3-Konsole ein Objekt herunterladen, dessen Schlüsselname mit Punkt (.) endet, wird der Punkt aus dem Schlüsselnamen des heruntergeladenen Objekts entfernt. Um den Punkt am Ende des Namens des heruntergeladenen Objekts beizubehalten, müssen Sie den AWS Command Line Interface (AWS CLI), AWS SDKs oder Amazon S3 RESTAPI.

Ein Objekt von einem S3-Bucket herunterladen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, aus dem Sie ein Objekt herunterladen möchten.
3. Sie können ein Objekt wie folgt aus einem S3-Bucket herunterladen:
 - Aktivieren Sie das Kontrollkästchen neben dem Objekt und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.
 - Wenn Sie eine spezifische Version des Objekts herunterladen möchten, aktivieren Sie die Schaltfläche Versionen anzeigen (neben dem Suchfeld). Aktivieren Sie das Kontrollkästchen neben der gewünschten Version des Objekts und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.

Verwendung der AWS CLI

Der folgende `get-object` Beispielbefehl zeigt, wie Sie den verwenden können AWS CLI um ein Objekt von Amazon S3 herunterzuladen. Mit diesem Befehl wird das Objekt `folder/my_image` aus dem Bucket `amzn-s3-demo-bucket1` abgerufen. Das Objekt wird in eine Datei mit dem Namen `my_downloaded_image` heruntergeladen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/  
my_image my_downloaded_image
```

Weitere Informationen und Beispiele finden Sie unter [get-object](#) in der AWS CLI Befehlsreferenz.

Verwendung der AWS SDKs

Beispiele für das Herunterladen eines Objekts mit dem AWS SDKs, siehe [Verwenden Sie GetObject mit einem AWS SDK oder CLI](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Mit dem REST API

Sie können den verwenden RESTAPI, um Objekte aus Amazon S3 abzurufen. Weitere Informationen finden Sie unter [GetObject](#) in der Amazon Simple Storage Service API Reference.

Herunterladen mehrerer Objekte

Sie können mehrere Objekte herunterladen, indem Sie AWS CloudShell, der AWS CLI, oder der AWS SDKs.

Die Verwendung von AWS CloudShell in der AWS Management Console

AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der AWS Management Console.

Weitere Informationen zur AWS CloudShell, [siehe Was ist? CloudShell](#) in der AWS CloudShell Benutzerleitfaden.

Important

Mit AWS CloudShell, Ihr Home-Verzeichnis hat Speicherplatz von bis zu 1 GB pro AWS-Region. Daher können Sie Buckets nicht mit Objekten synchronisieren, deren Gesamtwert diesen Betrag übersteigt. Weitere Einschränkungen finden Sie unter [Dienstkontingente und Einschränkungen](#) im AWS CloudShell Benutzerleitfaden.

Um Objekte herunterzuladen, verwenden Sie AWS CloudShell

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die CloudShell Konsole unter <https://console.aws.amazon.com/cloudshell/>.
2. Führen Sie den folgenden Befehl aus, um Objekte in Ihrem Bucket mit zu synchronisieren CloudShell. Der folgende Befehl synchronisiert Objekte aus dem Bucket mit dem Namen *amzn-s3-demo-bucket1* und erstellt einen Ordner mit dem Namen *temp* in CloudShell. CloudShell synchronisiert Ihre Objekte mit diesem Ordner. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws s3 sync s3://amzn-s3-demo-bucket1 ./temp
```

 Note

Wenn Sie einen Musterabgleich durchführen möchten, um bestimmte Objekte aus- oder einzuschließen, können Sie die Parameter `--exclude "value"` und `--include "value"` mit dem `sync`-Befehl verwenden.

3. Führen Sie den folgenden Befehl aus, um Ihre Objekte im Ordner `temp` in eine Datei mit dem Namen `temp.zip` zu komprimieren.

```
zip temp.zip -r temp/
```

4. Wählen Sie Aktionen und anschließend Datei herunterladen aus.
5. Geben Sie den Dateinamen `temp.zip` ein und wählen Sie anschließend Herunterladen aus.
6. (Optional) Löschen Sie die `temp.zip` Datei und die Objekte, die mit dem `temp` Ordner in synchronisiert wurden. CloudShell Mit AWS CloudShell, Sie haben persistenten Speicher von jeweils bis zu 1 GB AWS-Region.

Sie können den folgenden Beispielbefehl verwenden, um die `.zip`-Datei und den Ordner zu löschen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie `user input placeholders` durch Ihre Informationen.

```
rm temp.zip && rm -rf temp/
```

Verwendung der AWS CLI

Das folgende Beispiel zeigt, wie Sie das verwenden können AWS CLI um alle Dateien oder Objekte unter dem angegebenen Verzeichnis oder Präfix herunterzuladen. Mit diesem Befehl werden alle Objekte aus dem Bucket `amzn-s3-demo-bucket1` in Ihr aktuelles Verzeichnis kopiert. Wenn Sie diesen Beispielbefehl verwenden möchten, verwenden Sie anstelle von `amzn-s3-demo-bucket1` den Namen Ihres Buckets.

```
aws s3 cp s3://amzn-s3-demo-bucket1 . --recursive
```

Mit dem folgenden Befehl werden alle Objekte unter dem Präfix `logs` im Bucket `amzn-s3-demo-bucket1` in Ihr aktuelles Verzeichnis heruntergeladen. Darüber hinaus werden die Parameter `--exclude` und `--include` verwendet, damit nur Objekte mit dem Suffix `.log` kopiert werden. Wenn

Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3 cp s3://amzn-s3-demo-bucket1/logs/ . --recursive --exclude "*" --include "*.log"
```

Weitere Informationen und Beispiele finden Sie unter [cp](#) in der AWS CLI Befehlsreferenz.

Verwendung der AWS SDKs

Beispiele für das Herunterladen aller Objekte in einem Amazon S3 S3-Bucket mit dem AWS SDKs, siehe [Herunterladen aller Objekte aus einem Amazon Simple Storage Service \(Amazon S3\)-Bucket in ein lokales Verzeichnis](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Herunterladen eines Teils eines Objekts

Sie können einen Teil eines Objekts herunterladen, indem Sie AWS CLI oder RESTAPI. Dazu geben Sie mithilfe zusätzlicher Parameter an, welcher Teil eines Objekts heruntergeladen werden soll.

Verwendung der AWS CLI

Der folgende Beispielbefehl führt eine GET-Anforderung für einen Bytebereich im Objekt *folder/my_data* im Bucket *amzn-s3-demo-bucket1* aus. In der Anforderung muss dem Bytebereich das Präfix `bytes=` vorangestellt werden. Das Teilobjekt wird in die Ausgabedatei mit dem Namen *my_data_range* heruntergeladen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/my_data --range  
bytes=0-500 my_data_range
```

Weitere Informationen und Beispiele finden Sie unter [get-object](#) in der AWS CLI Befehlsreferenz.

Weitere Informationen zum HTTP Range Header finden Sie unter [RFC9110](#) auf der RFC Editor-Website.

Note

Amazon S3 unterstützt nicht das Abrufen mehrerer Datenbereiche in einer einzelnen GET-Anforderung.

Unter Verwendung des REST API

Sie können die Range Parameter `partNumber` und `in` verwenden RESTAPI, um Objektteile aus Amazon S3 abzurufen. Weitere Informationen finden Sie unter [GetObject](#) in der Amazon Simple Storage Service API Reference.

Ein Objekt von einem anderen herunterladen AWS-Konto

Sie können ein vorsigniertes Objekt verwenden URL, um anderen zeitlich begrenzten Zugriff auf Ihre Objekte zu gewähren, ohne Ihre Bucket-Richtlinie zu aktualisieren.

Das Vorsignierte URL kann in einem Browser eingegeben oder von einem Programm zum Herunterladen eines Objekts verwendet werden. Die von der verwendeten Anmeldeinformationen URL sind die von AWS Benutzer, der das generiert hat URL. Nachdem URL das erstellt wurde, URL kann jeder, der das vorsignierte Objekt besitzt, das entsprechende Objekt herunterladen, bis es URL abläuft.

Verwenden eines in der S3-Konsole URL vorsignierten

Sie können die Amazon S3 S3-Konsole verwenden, um ein Objekt zu generieren, das URL für die gemeinsame Nutzung vorsigniert ist, indem Sie die folgenden Schritte ausführen. Wenn Sie die Konsole verwenden, beträgt die maximale Ablaufzeit für ein vorsigniertes URL Objekt 12 Stunden ab dem Zeitpunkt der Erstellung.

Um mit der Amazon S3 S3-Konsole ein URL vorsigniertes zu generieren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Bucket-Liste den Namen des Buckets aus, der das Objekt enthält, für das Sie eine URL Vorsignierung wünschen.
4. Wählen Sie in der Objektliste das Objekt aus, für das Sie ein URL vorsigniertes Objekt erstellen möchten.

5. Wählen Sie im Menü „Objektaktionen“ die Option „Mit einem URL vorsignierten Objekt teilen“.
6. Geben Sie an, wie lange das Vorsignierte gültig sein URL soll.
7. Wählen Sie „URLVorsigniertes erstellen“.
8. Wenn eine Bestätigungsmeldung angezeigt wird, wird diese automatisch in Ihre Zwischenablage kopiert. URL Sie werden eine Schaltfläche sehen, mit der Sie das Vorsignierte kopieren könnenURL, falls Sie es erneut kopieren müssen.
9. Um das Objekt herunterzuladen, fügen Sie es URL in einen beliebigen Browser ein. Das Objekt versucht dann, es herunterzuladen.

Weitere Hinweise zu vorsignierten URLs und anderen Methoden zu ihrer Erstellung finden Sie unter [Sie arbeiten mit vorsignierten URLs](#).

Herunterladen von archivierten Objekten

Sie können Objekte, auf die selten zugegriffen wird, archivieren, um Ihre Speicherkosten dafür zu senken. Wenn Sie ein Objekt archivieren, wird es in einen kostengünstigen Speicher verschoben. Sie können also nicht in Echtzeit darauf zugreifen. Wenn Sie ein archiviertes Objekt herunterladen möchten, müssen Sie es zunächst wiederherstellen.

Je nach Speicherklasse können archivierte Objekte innerhalb von Minuten oder Stunden wiederhergestellt werden. Sie können ein archiviertes Objekt mithilfe der Amazon S3-Konsole, S3 Batch Operations, Amazon S3 RESTAPI, der AWS SDKs, und die AWS Command Line Interface (AWS CLI).

Detaillierte Anweisungen finden Sie unter [Wiederherstellen eines archivierten Objekts](#). Nachdem Sie das archivierte Objekt wiederhergestellt haben, können Sie es herunterladen.

Objekte auf der Grundlage von Metadaten herunterladen

Sie können Vorbedingungen hinzufügen, um ein Objekt anhand seiner Metadaten mithilfe einer bedingten Leseanforderung herunterzuladen. Sie können ein Objekt auf der Grundlage seines Entity-Tags (ETag) oder des Datums der letzten Änderung zurückgeben. Dies kann eine S3-Operation auf Objekte beschränken, die seit einem bestimmten Datum aktualisiert wurden, oder nur eine bestimmte Objektversion zurückgeben.

Sie können bedingte Schreibvorgänge verwenden für [GetObject](#) oder [HeadObject](#)Anfragen.

Weitere Informationen zu bedingten Anfragen finden Sie unter [Fügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu](#).

Fehlerbehebung beim Herunterladen von Objekten

Unzureichende Berechtigungen oder falscher Bucket oder AWS Identity and Access Management (IAM) Benutzerrichtlinien können zu Fehlern führen, wenn Sie versuchen, Objekte von Amazon S3 herunterzuladen. Diese Probleme können häufig den Fehler Zugriff verweigert (403 Verboten) hervorrufen. Dies bedeutet, dass Amazon S3 den Zugriff auf eine Ressource nicht gewähren kann.

Weitere Informationen zu den häufigsten Ursachen für den Fehler Zugriff verweigert (403 Verboten) finden Sie unter [Problembehandlung bei Zugriff verweigert \(403 Forbidden\) Fehler in Amazon S3](#).

Überprüfung der Objektintegrität

Amazon S3 verwendet Prüfsummenwerte, um die Integrität von Daten zu verifizieren, die Sie auf Amazon S3 hochladen oder von Amazon S3 herunterladen. Darüber hinaus können Sie anfordern, dass ein weiterer Prüfsummenwert für jedes Objekt berechnet wird, das Sie in Amazon S3 speichern. Sie können einen von mehreren Prüfsummenalgorithmen auswählen, der beim Hochladen oder Kopieren Ihrer Daten verwendet werden soll. Amazon S3 berechnet mit diesem Algorithmus einen zusätzlichen Prüfsummenwert und speichert ihn als Teil der Objektmetadaten. Weitere Informationen zur Verwendung zusätzlicher Prüfsummen zur Überprüfung der Datenintegrität finden Sie im [Tutorial: Überprüfen der Integrität von Daten in Amazon S3 mit zusätzlichen Prüfsummen](#).

Wenn Sie ein Objekt hochladen, können Sie optional eine vorberechnete Prüfsumme als Teil Ihrer Anforderung aufnehmen. Amazon S3 vergleicht die bereitgestellte Prüfsumme mit der Prüfsumme, die mit Ihrem angegebenen Algorithmus berechnet wird. Wenn die beiden Werte nicht übereinstimmen, meldet Amazon S3 einen Fehler.

Verwenden unterstützter Prüfsummenalgorithmen

Amazon S3 bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) wählen:

- CRC-32
- CRC-32 °C


- SHA-1
- SHA-256

Wenn Sie ein Objekt hochladen, können Sie den Algorithmus angeben, den Sie verwenden möchten:

- Wenn du das benutzt AWS Management Console, wählen Sie den Prüfsummenalgorithmus aus, den Sie verwenden möchten. In diesem Fall können Sie den Prüfsummenwert des Objekts optional angeben. Wenn Amazon S3 das Objekt erhält, berechnet es die Prüfsumme mithilfe des von Ihnen angegebenen Algorithmus. Wenn die beiden Prüfsummenwerte nicht übereinstimmen, generiert Amazon S3 einen Fehler.
- Wenn Sie einen verwenden SDK, können Sie den Wert des `ChecksumAlgorithm` Parameters auf den Algorithmus setzen, den Amazon S3 bei der Berechnung der Prüfsumme verwenden soll. Amazon S3 berechnet den Prüfsummenwert automatisch.
- Wenn Sie den verwenden REST API, verwenden Sie den `x-amz-sdk-checksum-algorithm` Parameter nicht. Stattdessen verwenden Sie einen der algorithmenspezifischen Header (z. B. `x-amz-checksum-crc32`).

Weitere Informationen zum Hochladen von Objekten finden Sie unter [Objekte hochladen](#).

Um einen dieser Prüfsummenwerte auf Objekte anzuwenden, die bereits auf Amazon S3 hochgeladen wurden, können Sie das Objekt kopieren. Wenn Sie ein Objekt kopieren, können Sie angeben, ob Sie den vorhandenen oder einen neuen Prüfsummenalgorithmus verwenden möchten. Sie können einen Prüfsummenalgorithmus angeben, wenn Sie eine unterstützte Methode zum Kopieren von Objekten verwenden, einschließlich S3-Batch-Operationen. Weitere Informationen über S3-Batch-Vorgänge finden Sie unter [Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch](#).

 **Important**

Wenn Sie einen mehrteiligen Upload mit zusätzlichen Prüfsummen verwenden, müssen die mehrteiligen Teilenummern aufeinander folgende Teilenummern sein. Wenn Sie zusätzliche Prüfsummen verwenden und versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen Fehler. `HTTP 500 Internal Server Error`

Nach dem Hochladen von Objekten können Sie den Prüfsummenwert abrufen und mit einem vorberechneten oder zuvor gespeicherten Prüfsummenwert vergleichen, der mit demselben Algorithmus berechnet wurde.

Verwenden der S3-Konsole

Weitere Informationen zur Verwendung der Konsole und zum Angeben von Prüfsummenalgorithmen, die beim Hochladen von Objekten verwendet werden, finden Sie unter [Objekte hochladen](#) und unter [Tutorial: Überprüfen der Integrität von Daten in Amazon S3 mit zusätzlichen Prüfsummen](#).

Verwendung der AWS SDKs

Das folgende Beispiel zeigt, wie Sie AWS SDKsum eine große Datei mit mehrteiligem Upload hochzuladen, eine große Datei herunterzuladen und eine mehrteilige Upload-Datei zu validieren, alles mit SHA -256 für die Dateiüberprüfung.

Java

Example Beispiel: Hochladen, Herunterladen und Überprüfen einer großen Datei mit -256 SHA

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import software.amazon.awssdk.auth.credentials.AwsCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AbortMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesRequest;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
```



```
import software.amazon.awssdk.services.s3.model.ObjectAttributes;
import software.amazon.awssdk.services.s3.model.PutObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.Tag;
import software.amazon.awssdk.services.s3.model.Tagging;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Base64;
import java.util.List;

public class LargeObjectValidation {
    private static String FILE_NAME = "sample.file";
    private static String BUCKET = "sample-bucket";
    //Optional, if you want a method of storing the full multipart object
checksum in S3.
    private static String CHECKSUM_TAG_KEYNAME = "fullObjectChecksum";
    //If you have existing full-object checksums that you need to validate
against, you can do the full object validation on a sequential upload.
    private static String SHA256_FILE_BYTES = "htCM5g7ZNdoSw8bN/
mkgiAhXt5MFoVowVg+LE9aIQmI=";
    //Example Chunk Size - this must be greater than or equal to 5MB.
    private static int CHUNK_SIZE = 5 * 1024 * 1024;

    public static void main(String[] args) {
        S3Client s3Client = S3Client.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(new AwsCredentialsProvider() {
                @Override
                public AwsCredentials resolveCredentials() {
                    return new AwsCredentials() {
                        @Override
                        public String accessKeyId() {
                            return Constants.ACCESS_KEY;
                        }
                    };
                }
            })
    }
```

```

        @Override
        public String secretAccessKey() {
            return Constants.SECRET;
        }
    };
}
}
}
    .build();
uploadLargeFileBracketedByChecksum(s3Client);
downloadLargeFileBracketedByChecksum(s3Client);
validateExistingFileAgainstS3Checksum(s3Client);
}

public static void uploadLargeFileBracketedByChecksum(S3Client s3Client) {
    System.out.println("Starting uploading file validation");
    File file = new File(FILE_NAME);
    try (InputStream in = new FileInputStream(file)) {
        MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
        CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
        .bucket(BUCKET)
        .key(FILE_NAME)
        .checksumAlgorithm(ChecksumAlgorithm.SHA256)
        .build();
        CreateMultipartUploadResponse createdUpload =
s3Client.createMultipartUpload(createMultipartUploadRequest);
        List<CompletedPart> completedParts = new ArrayList<CompletedPart>();
        int partNumber = 1;
        byte[] buffer = new byte[CHUNK_SIZE];
        int read = in.read(buffer);
        while (read != -1) {
            UploadPartRequest uploadPartRequest =
UploadPartRequest.builder()
                .partNumber(partNumber).uploadId(createdUpload.uploadId()).key(FILE_NAME).bucket(BUCKET).ch
                UploadPartResponse uploadedPart =
s3Client.uploadPart(uploadPartRequest,
RequestBody.fromByteBuffer(ByteBuffer.wrap(buffer, 0, read)));
            CompletedPart part =
CompletedPart.builder().partNumber(partNumber).checksumSHA256(uploadedPart.checksumSHA256())
                completedParts.add(part);
            sha256.update(buffer, 0, read);
            read = in.read(buffer);

```

```

        partNumber++;
    }
    String fullObjectChecksum =
Base64.getEncoder().encodeToString(sha256.digest());
    if (!fullObjectChecksum.equals(SHA256_FILE_BYTES)) {
        //Because the SHA256 is uploaded after the part is uploaded; the
upload is bracketed and the full object can be fully validated.

s3Client.abortMultipartUpload(AbortMultipartUploadRequest.builder().bucket(BUCKET).key(FILE
        throw new IOException("Byte mismatch between stored checksum and
upload, do not proceed with upload and cleanup");
    }
    CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder().parts(completedParts).build();
    CompleteMultipartUploadResponse completedUploadResponse =
s3Client.completeMultipartUpload(

CompleteMultipartUploadRequest.builder().bucket(BUCKET).key(FILE_NAME).uploadId(createdUplo
        Tag checksumTag =
Tag.builder().key(CHECKSUM_TAG_KEYNAME).value(fullObjectChecksum).build();
        //Optionally, if you need the full object checksum stored with the
file; you could add it as a tag after completion.

s3Client.putObjectTagging(PutObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).t
    } catch (IOException | NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    GetObjectAttributesResponse
        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
        System.out.println(objectAttributes.objectParts().parts());
        System.out.println(objectAttributes.checksum().checksumSHA256());
    }

    public static void downloadLargeFileBracketedByChecksum(S3Client s3Client) {
        System.out.println("Starting downloading file validation");
        File file = new File("DOWNLOADED_" + FILE_NAME);
        try (OutputStream out = new FileOutputStream(file)) {
            GetObjectAttributesResponse
                objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N

```

```

        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
        //Optionally if you need the full object checksum, you can grab a
tag you added on the upload
        List<Tag> objectTags =
s3Client.getObjectTagging(GetObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).b
        String fullObjectChecksum = null;
        for (Tag objectTag : objectTags) {
            if (objectTag.key().equals(CHECKSUM_TAG_KEYNAME)) {
                fullObjectChecksum = objectTag.value();
                break;
            }
        }
        MessageDigest sha256FullObject =
MessageDigest.getInstance("SHA-256");
        MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");

        //If you retrieve the object in parts, and set the ChecksumMode to
enabled, the SDK will automatically validate the part checksum
        for (int partNumber = 1; partNumber <=
objectAttributes.objectParts().totalPartsCount(); partNumber++) {
            MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
            ResponseInputStream<GetObjectResponse> response =
s3Client.getObject(GetObjectRequest.builder().bucket(BUCKET).key(FILE_NAME).partNumber(part
            GetObjectResponse getObjectResponse = response.response();
            byte[] buffer = new byte[CHUNK_SIZE];
            int read = response.read(buffer);
            while (read != -1) {
                out.write(buffer, 0, read);
                sha256FullObject.update(buffer, 0, read);
                sha256Part.update(buffer, 0, read);
                read = response.read(buffer);
            }
            byte[] sha256PartBytes = sha256Part.digest();
            sha256ChecksumOfChecksums.update(sha256PartBytes);
            //Optionally, you can do an additional manual validation again
the part checksum if needed in addition to the SDK check
            String base64PartChecksum =
Base64.getEncoder().encodeToString(sha256PartBytes);
            String base64PartChecksumFromObjectAttributes =
objectAttributes.objectParts().parts().get(partNumber - 1).checksumSHA256();

```

```

        if (!
base64PartChecksum.equals(getObjectResponse.checksumSHA256()) || !
base64PartChecksum.equals(base64PartChecksumFromObjectAttributes)) {
            throw new IOException("Part checksum didn't match for the
part");
        }
        System.out.println(partNumber + " " + base64PartChecksum);
    }
    //Before finalizing, do the final checksum validation.
    String base64FullObject =
Base64.getEncoder().encodeToString(sha256FullObject.digest());
    String base64ChecksumOfChecksums =
Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
    if (fullObjectChecksum != null && !
fullObjectChecksum.equals(base64FullObject)) {
        throw new IOException("Failed checksum validation for full
object");
    }
    System.out.println(fullObjectChecksum);
    String base64ChecksumOfChecksumFromAttributes =
objectAttributes.checksum().checksumSHA256();
    if (base64ChecksumOfChecksumFromAttributes != null && !
base64ChecksumOfChecksums.equals(base64ChecksumOfChecksumFromAttributes)) {
        throw new IOException("Failed checksum validation for full
object checksum of checksums");
    }
    System.out.println(base64ChecksumOfChecksumFromAttributes);
    out.flush();
} catch (IOException | NoSuchAlgorithmException e) {
    //Cleanup bad file
    file.delete();
    e.printStackTrace();
}
}

public static void validateExistingFileAgainstS3Checksum(S3Client s3Client)
{
    System.out.println("Starting existing file validation");
    File file = new File("DOWNLOADED_" + FILE_NAME);
    GetObjectAttributesResponse
        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());

```

```

        try (InputStream in = new FileInputStream(file)) {
            MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");
            MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
            byte[] buffer = new byte[CHUNK_SIZE];
            int currentPart = 0;
            int partBreak =
objectAttributes.objectParts().parts().get(currentPart).size();
            int totalRead = 0;
            int read = in.read(buffer);
            while (read != -1) {
                totalRead += read;
                if (totalRead >= partBreak) {
                    int difference = totalRead - partBreak;
                    byte[] partChecksum;
                    if (totalRead != partBreak) {
                        sha256Part.update(buffer, 0, read - difference);
                        partChecksum = sha256Part.digest();
                        sha256ChecksumOfChecksums.update(partChecksum);
                        sha256Part.reset();
                        sha256Part.update(buffer, read - difference,
difference);
                    } else {
                        sha256Part.update(buffer, 0, read);
                        partChecksum = sha256Part.digest();
                        sha256ChecksumOfChecksums.update(partChecksum);
                        sha256Part.reset();
                    }
                    String base64PartChecksum =
Base64.getEncoder().encodeToString(partChecksum);
                    if (!
base64PartChecksum.equals(objectAttributes.objectParts().parts().get(currentPart).checksumSH
{
                        throw new IOException("Part checksum didn't match S3");
                    }
                    currentPart++;
                    System.out.println(currentPart + " " + base64PartChecksum);
                    if (currentPart <
objectAttributes.objectParts().totalPartsCount()) {
                        partBreak +=
objectAttributes.objectParts().parts().get(currentPart - 1).size();
                    }
                } else {
                    sha256Part.update(buffer, 0, read);

```

```

        }
        read = in.read(buffer);
    }
    if (currentPart != objectAttributes.objectParts().totalPartsCount())
    {
        currentPart++;
        byte[] partChecksum = sha256Part.digest();
        sha256ChecksumOfChecksums.update(partChecksum);
        String base64PartChecksum =
Base64.getEncoder().encodeToString(partChecksum);
        System.out.println(currentPart + " " + base64PartChecksum);
    }

    String base64CalculatedChecksumOfChecksums =
Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
    System.out.println(base64CalculatedChecksumOfChecksums);
    System.out.println(objectAttributes.checksum().checksumSHA256());
    if (!
base64CalculatedChecksumOfChecksums.equals(objectAttributes.checksum().checksumSHA256()))
    {
        throw new IOException("Full object checksum of checksums don't
match S3");
    }

    } catch (IOException | NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
}
}
}

```

Unter Verwendung des REST API

Sie können REST Anfragen zum Hochladen eines Objekts mit einem Prüfsummenwert senden, mit [PutObject](#) dem die Integrität der Daten überprüft werden kann. Sie können den Prüfsummenwert für Objekte auch mit [GetObject](#) oder abrufen. [HeadObject](#)

Verwendung der AWS CLI

Sie können eine PUT-Anforderung zum Hochladen eines Objekts von bis zu 5 GB in einem einzigen Vorgang senden. Weitere Informationen finden Sie [PutObject](#) im AWS CLI Befehlsreferenz. Sie können auch [get-object](#) und [head-object](#) verwenden, um die Prüfsumme eines bereits hochgeladenen Objekts abzurufen und die Integrität der Daten zu überprüfen.

Weitere Informationen finden Sie unter [Amazon S3 CLI FAQ](#) in der AWS Command Line Interface Benutzerleitfaden.

Inhalte verwenden — MD5 beim Hochladen von Objekten

Eine weitere Möglichkeit, die Integrität Ihres Objekts nach dem Hochladen zu überprüfen, besteht darin, MD5 beim Hochladen eine Zusammenfassung des Objekts bereitzustellen. Wenn Sie den MD5 Digest für Ihr Objekt berechnen, können Sie den Digest mit dem PUT Befehl bereitstellen, indem Sie den Header verwenden. Content-MD5

Nach dem Hochladen des Objekts berechnet Amazon S3 den MD5 Digest des Objekts und vergleicht ihn mit dem von Ihnen angegebenen Wert. Die Anforderung ist nur erfolgreich, wenn die beiden Digests übereinstimmen.

Die Angabe eines MD5 Digest ist nicht erforderlich, Sie können ihn jedoch verwenden, um die Integrität des Objekts im Rahmen des Upload-Vorgangs zu überprüfen.

Verwenden Sie Content-MD5 und the, um hochgeladene ETag Objekte zu verifizieren

Das Entity-Tag (ETag) für ein Objekt steht für eine bestimmte Version dieses Objekts. Beachten Sie, dass das nur Änderungen am Inhalt eines Objekts ETag widerspiegelt, nicht an seinen Metadaten. Wenn sich nur die Metadaten eines Objekts ändern, ETag bleiben die gleichen.

Je nach Objekt kann es sich ETag bei dem Objekt um eine MD5 Zusammenfassung der Objektdaten handeln:

- Wenn ein Objekt durch die CopyObject Operation PutObjectPostObject, oder oder durch die erstellt wird AWS Management Console, und dieses Objekt ist auch Klartext oder durch serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt, ETag dieses Objekt hat eine MD5 Zusammenfassung seiner Objektdaten.
- Wenn ein Objekt durch die CopyObject Operation,, oder erstellt wird PutObjectPostObject, oder durch AWS Management Console, und dieses Objekt wird durch serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) oder serverseitige Verschlüsselung mit verschlüsselt AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), das Objekt hat eine, bei der es ETag sich nicht um eine MD5 Zusammenfassung seiner Objektdaten handelt.
- Wenn ein Objekt entweder durch den mehrteiligen Upload-Prozess oder durch den UploadPartCopy Vorgang erstellt wird, ETag handelt es sich bei dem Objekt unabhängig von der

Verschlüsselungsmethode nicht um einen MD5 Digest. Wenn ein Objekt größer als 16 MB ist, AWS Management Console lädt das Objekt als mehrteiligen Upload hoch oder kopiert es und ist daher ETag kein MD5 Digest.

Bei Objekten, bei denen ETag es sich um den Content-MD5 Digest des Objekts handelt, können Sie den ETag Wert des Objekts mit einem berechneten oder zuvor gespeicherten Digest vergleichen.
Content-MD5

Verwenden von nachfolgenden Prüfsummen

Wenn Sie Objekte auf Amazon S3 hochladen, können Sie entweder eine vorberechnete Prüfsumme für das Objekt angeben oder eine AWS SDK um automatisch abschließende Prüfsummen in Ihrem Namen zu erstellen. Wenn Sie sich für eine nachfolgende Prüfsumme entscheiden, generiert Amazon S3 die Prüfsumme automatisch mithilfe des von Ihnen angegebenen Algorithmus und überprüft damit die Integrität des Objekts während des Uploads.

Um eine abschließende Prüfsumme zu erstellen, wenn Sie ein AWS SDK, füllen Sie den `ChecksumAlgorithm` Parameter mit Ihrem bevorzugten Algorithmus. Der SDK verwendet diesen Algorithmus, um die Prüfsumme für Ihr Objekt (oder Objektteile) zu berechnen, und hängt sie automatisch an das Ende Ihrer Upload-Anfrage an. Dank dieser Funktionsweise sparen Sie Zeit, da Amazon S3 sowohl die Überprüfung als auch das Hochladen Ihrer Daten in einem einzigen Durchgang durchführt.

Important

Wenn Sie S3 Object Lambda verwenden, werden alle Anfragen an S3 Object Lambda mit `s3-object-lambda` anstelle von `s3` signiert. Dieses Verhalten wirkt sich auf die Signatur der nachfolgenden Prüfsummenwerte aus. Weitere Informationen zu S3 Object Lambda finden Sie unter [Transformieren von Objekten mit S3 Object Lambda](#).

Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads

Wenn Objekte auf Amazon S3 hochgeladen werden, können sie entweder als einzelnes Objekt oder durch den mehrteiligen Upload-Prozess hochgeladen werden. Objekte, die größer als 16 MB sind und über die Konsole hochgeladen werden, werden automatisch mit mehrteiligen Uploads hochgeladen. Weitere Informationen über mehrteilige Uploads finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Wenn ein Objekt als mehrteiliger Upload hochgeladen wird, handelt es sich ETag bei dem Objekt nicht um eine MD5 Zusammenfassung des gesamten Objekts. Amazon S3 berechnet den MD5 Digest jedes einzelnen Teils beim Hochladen. Die MD5 Digests werden verwendet, um den Wert ETag für das endgültige Objekt zu bestimmen. Amazon S3 verkettet die Bytes für die MD5 Digests miteinander und berechnet dann den Digest dieser verketteten MD5 Werte. Der letzte Schritt bei der Erstellung von ETag besteht darin, dass Amazon S3 am Ende einen Bindestrich mit der Gesamtzahl der Teile hinzufügt.


Stellen Sie sich zum Beispiel ein Objekt vor, das mit einem mehrteiligen Upload hochgeladen wurde und den Wert ETag von C9A5A6878D97B48CC965C1E41859F034-14 hat. In diesem Fall C9A5A6878D97B48CC965C1E41859F034 ist der MD5 Digest aller Digests miteinander verkettet. Der Wert -14 gibt an, dass mit dem mehrteiligen Upload dieses Objekts 14 Teile verknüpft sind.

Wenn Sie zusätzliche Prüfsummenwerte für Ihr mehrteiliges Objekt aktiviert haben, berechnet Amazon S3 die Prüfsumme für jeden einzelnen Teil mithilfe des angegebenen Prüfsummenalgorithmus. Die Prüfsumme für das abgeschlossene Objekt wird auf die gleiche Weise berechnet, wie Amazon S3 den MD5 Digest für den mehrteiligen Upload berechnet. Mithilfe dieser Prüfsumme können Sie die Integrität des Objekts überprüfen.


Um Informationen über das Objekt abzurufen, einschließlich der Anzahl der Teile, aus denen das gesamte Objekt besteht, können Sie die Operation verwenden. [GetObjectAttributes](#) Mit zusätzlichen Prüfsummen können Sie auch Informationen für jeden einzelnen Teil wiederherstellen, der den Prüfsummenwert jedes Teils enthält.

Bei abgeschlossenen Uploads können Sie die Prüfsumme eines einzelnen Teils abrufen, indem Sie die [HeadObject](#) Operationen [GetObject](#) oder verwenden und eine Bauteilnummer oder einen Bytebereich angeben, der einem einzelnen Teil entspricht. Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie Folgendes verwenden. [ListParts](#)

Aufgrund der Art und Weise, wie Amazon S3 die Prüfsumme für mehrteilige Objekte berechnet, kann sich der Prüfsummenwert für das Objekt ändern, wenn Sie es kopieren. Wenn Sie ein SDK oder das verwenden REST API und Sie aufrufen [CopyObject](#), kopiert Amazon S3 jedes Objekt bis zu den Größenbeschränkungen des CopyObject API Vorgangs. Amazon S3 führt diese Kopie als einzelne Aktion aus, unabhängig davon, ob das Objekt in einer einzigen Anforderung oder im Rahmen eines mehrteiligen Uploads hochgeladen wurde. Mit einem Kopierbefehl ist die Prüfsumme des Objekts eine direkte Prüfsumme des vollständigen Objekts. Wenn das Objekt ursprünglich mit einem mehrteiligen Upload hochgeladen wurde, ändert sich der Prüfsummenwert, auch wenn die Daten unverändert bleiben.

 Note

Objekte, die die Größenbeschränkungen des CopyObject API Vorgangs überschreiten, müssen mehrteilige Kopierbefehle verwenden.

 Important

Wenn Sie einige Operationen mit dem AWS Management Console, Amazon S3 verwendet einen mehrteiligen Upload, wenn das Objekt größer als 16 MB ist. In diesem Fall ist die Prüfsumme keine direkte Prüfsumme des vollständigen Objekts, sondern eine Berechnung, die auf den Prüfsummenwerten jeden einzelnen Teils basiert.

Stellen Sie sich zum Beispiel ein Objekt mit einer Größe von 100 MB vor, das Sie als einteiligen direkten Upload mit dem hochgeladen haben. REST API Die Prüfsumme ist in diesem Fall eine Prüfsumme des gesamten Objekts. Wenn Sie das Objekt später mit der Konsole umbenennen, es kopieren, die Speicherklasse ändern oder die Metadaten bearbeiten, verwendet Amazon S3 die mehrteilige Upload-Funktion, um das Objekt zu aktualisieren. Infolgedessen erstellt Amazon S3 einen neuen Prüfsummenwert für das Objekt, der basierend auf den Prüfsummenwerten der einzelnen Teile berechnet wird.

Die obige Liste der Konsolenoperationen ist keine vollständige Liste aller möglichen Aktionen, die Sie in der AWS Management Console die dazu führen, dass Amazon S3 das Objekt mithilfe der mehrteiligen Upload-Funktion aktualisiert. Beachten Sie, dass der Prüfsummenwert möglicherweise nicht die Prüfsumme des gesamten Objekts ist, wenn Sie die Konsole für Objekte mit einer Größe über 16 MB verwenden.

Löschen von Amazon-S3-Objekten

Sie können mit der Amazon S3-Konsole ein oder mehrere Objekte direkt aus Amazon S3 löschen. AWS SDKs, AWS Command Line Interface (AWS CLI), oder RESTAPI. Für alle Objekte, die Sie in Ihrem S3-Bucket aufbewahren, entstehen Speicherkosten, deshalb sollten Sie Objekte löschen, die Sie nicht mehr brauchen. Wenn Sie beispielsweise Protokolldateien sammeln, sollten Sie sie unbedingt löschen, wenn sie nicht mehr benötigt werden. Sie können einen Lebenszyklusregel definieren, um Objekte wie Protokolldateien automatisch zu löschen. Weitere Informationen finden Sie unter [the section called "Einrichten der Lebenszyklus-Konfiguration"](#).

Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon-S3-Preise](#).

Beim Löschen eines Objekts haben Sie die folgenden API Optionen:

- Ein einzelnes Objekt löschen — Amazon S3 bietet die API Operation DELETE (DeleteObject), mit der Sie ein Objekt in einer einzigen HTTP Anfrage löschen können.
- Mehrere Objekte löschen — Amazon S3 bietet den Multi-Object Delete (DeleteObjects) API - Vorgang, mit dem Sie bis zu 1.000 Objekte in einer einzigen HTTP Anfrage löschen können.

Wenn Sie Objekte aus einem Bucket löschen, in dem die Versionsverwaltung nicht aktiviert ist, geben Sie nur den Namen des Objektschlüssels an. Beim Löschen von Objekten aus einem Bucket mit aktivierter Versionierung können Sie jedoch optional die Versions-ID des Objekts angeben, um eine bestimmte Version des Objekts zu löschen.

Programmgesteuertes Löschen von Objekten aus einem Bucket mit aktivierter Versionierung

Wenn für Ihren Bucket die Versionsverwaltung aktiviert ist, können mehrere Versionen desselben Objekts im Bucket vorhanden sein. Bei der Arbeit mit Buckets mit aktivierter Versionierung ermöglichen die API Löschvorgänge die folgenden Optionen:

- Eine Löschanforderung ohne Versions-ID angeben – Geben Sie nur den Schlüssel des Objekts an, nicht die Versions-ID. In diesem Fall erstellt Amazon S3 eine Löschkennzeichnung und gibt in der Antwort ihre Versions-ID zurück. Dadurch verschwindet Ihr Objekt aus dem Bucket. Weitere Informationen zum Objekt-Versioning und zum Konzept der Löschkennzeichnung finden Sie unter [Verwenden der Versioning in S3-Buckets](#).
- Eine Löschanforderung mit Versions-ID angeben– Geben Sie sowohl den Schlüssel als auch eine Versions-ID an. In diesem Fall sind zwei Ergebnisse möglich:
 - Wenn die Versions-ID einer spezifischen Objektversion zugeordnet ist, löscht Amazon S3 die spezifische Version des Objekts.
 - Wenn die Versions-ID der Löschkennzeichnung dieses Objekts zugeordnet ist, löscht Amazon S3 die Löschkennzeichnung. Dadurch wird Ihr Objekt wieder in Ihrem Bucket angezeigt.

Löschen von Objekten aus einem Bucket mit aktivierter Option MFA

Beachten Sie beim Löschen von Objekten aus einem Bucket mit aktivierter Multi-Faktor-Authentifizierung (MFA) Folgendes:

- Wenn Sie ein ungültiges MFA Token angeben, schlägt die Anfrage immer fehl.
- Wenn Sie einen Bucket mit MFA aktiviertem Bucket haben und eine versionierte Löschanforderung stellen (Sie geben einen Objektschlüssel und eine Versions-ID an), schlägt die Anfrage fehl, wenn Sie kein MFA gültiges Token angeben. Außerdem schlägt die gesamte Anforderung fehl, wenn Sie den API Vorgang zum Löschen mehrerer Objekte für einen Bucket mit MFA aktivierter Option verwenden und es sich bei einer der Löschungen um eine versionierte Löschanforderung handelt (d. h. Sie geben einen Objektschlüssel und eine Versions-ID an), wenn Sie kein Token angeben.
MFA

In den folgenden Fällen ist die Anfrage jedoch erfolgreich:

- Wenn Sie einen Bucket mit MFA aktiviertem Bucket haben und eine nicht versionierte Löschanforderung stellen (Sie löschen kein versioniertes Objekt) und Sie kein Token angeben, ist der Löschvorgang erfolgreich. MFA
- Wenn Sie eine Anfrage zum Löschen mehrerer Objekte haben, in der nur nicht versionierte Objekte aus einem Bucket mit MFA aktiviertem Bucket gelöscht werden sollen, und Sie kein Token angeben, sind die Löschungen erfolgreich. MFA

Informationen zum Löschen finden Sie unter. MFA [MFA Löschen konfigurieren](#)

Themen

- [Löschen eines einzelnen Objekts](#)
- [Löschen von mehreren Objekten](#)

Löschen eines einzelnen Objekts

Sie können die Amazon S3 S3-Konsole oder die verwenden DELETEAPI, um ein einzelnes vorhandenes Objekt aus einem S3-Bucket zu löschen. Weitere Informationen zum Löschen von Objekten in Amazon S3 finden Sie unter [Löschen von Amazon-S3-Objekten](#).

Für alle Objekte, die Sie in Ihrem S3-Bucket aufbewahren, entstehen Speicherkosten, deshalb sollten Sie Objekte löschen, die Sie nicht mehr brauchen. Wenn Sie beispielsweise Protokolldateien sammeln, sollten Sie sie unbedingt löschen, wenn sie nicht mehr benötigt werden. Sie können einen Lebenszyklusregel definieren, um Objekte wie Protokolldateien automatisch zu löschen. Weitere Informationen finden Sie unter [the section called “Einrichten der Lebenszyklus-Konfiguration”](#).

Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon-S3-Preise](#).

Verwenden der S3-Konsole

Befolgen Sie diese Schritte, um mithilfe der Amazon-S3-Konsole ein einzelnes Objekt aus einem Bucket zu löschen.

Warning

Wenn Sie ein Objekt oder eine bestimmte Objektversion dauerhaft in der Amazon S3 S3-Konsole löschen, kann der Löschvorgang nicht rückgängig gemacht werden.

Um ein Objekt zu löschen, für das die Versionsverwaltung aktiviert oder ausgesetzt wurde

Note

Wenn die Versions-ID für ein Objekt in einem Bucket, für das die Versionierung gesperrt wurde NULL, als markiert ist, löscht S3 das Objekt dauerhaft, da keine früheren Versionen existieren. Wenn jedoch eine gültige Versions-ID für das Objekt in einem Bucket aufgeführt ist, für das die Versionierung gesperrt wurde, erstellt S3 eine Löschmarkierung für das gelöschte Objekt, wobei die vorherigen Versionen des Objekts beibehalten werden.

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
3. Wählen Sie das Objekt aus und wählen Sie dann Löschen.
4. Um das Löschen der Objektliste unter Bestimmte Objekte im Bereich Objekte löschen zu bestätigen? Geben Sie in das Textfeld ein **delete**.

Um eine bestimmte Objektversion in einem Bucket mit aktivierter Versionierung dauerhaft zu löschen

Warning

Wenn Sie eine bestimmte Objektversion in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
3. Wählen Sie das zu löschende Objekt aus.
4. Wählen Sie den Schalter Versionen anzeigen.
5. Wählen Sie die Objektversion aus und klicken Sie dann auf Löschen.
6. Um das dauerhafte Löschen der spezifischen Objektversionen zu bestätigen, die unter Bestimmte Objekte im Bereich Objekte löschen? aufgeführt sind Geben Sie im Textfeld Dauerhaft löschen ein. Amazon S3 löscht die spezifische Objektversion dauerhaft.

Um ein Objekt in einem Amazon S3 S3-Bucket, für den die Versionierung nicht aktiviert ist, dauerhaft zu löschen

Warning

Wenn Sie ein Objekt in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden. Außerdem sind Löschungen für alle Buckets ohne aktivierte Versionierung dauerhaft.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
3. Wählen Sie das Objekt aus und wählen Sie dann Löschen.
4. Um das dauerhafte Löschen des Objekts zu bestätigen, das unter Bestimmte Objekte im Feld Objekte löschen? aufgeführt ist Geben Sie im Textfeld Dauerhaft löschen ein.

Note

Falls beim Löschen Ihres Objekts Probleme auftreten, finden Sie weitere Informationen unter [Lich möchte versionierte Objekte dauerhaft löschen](#).

Verwendung der AWS SDKs

Die folgenden Beispiele zeigen, wie Sie das verwenden können AWS SDKsum ein Objekt aus einem Bucket zu löschen. Weitere Informationen finden Sie unter [DELETEObjekt](#) in der Amazon Simple Storage Service API Reference

Wenn für den Bucket das S3-Versioning aktiviert ist, stehen Ihnen die folgenden Optionen zur Verfügung:

- Löschen einer spezifischen Objektversion durch Angabe einer Versions-ID.
- Löschen eines Objekts ohne Angabe einer Version-ID. In diesem Fall fügt Amazon S3 dem Objekt eine Löschmarkierung hinzu.

Weitere Informationen über das S3-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Java

Example Beispiel 1: Löschen eines Objekts (nicht versionierter Bucket)

Im folgenden Beispiel wird davon ausgegangen, dass der Bucket nicht versionierungsfähig ist und das Objekt keine Version hat. IDs In der Löschanfrage geben Sie nur den Objektschlüssel und keine Versions-ID an.

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
```



```
import com.amazonaws.services.s3.model.DeleteObjectRequest;

import java.io.IOException;

public class DeleteObjectNonVersionedBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();


            s3Client.deleteObject(new DeleteObjectRequest(bucketName, keyName));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Beispiel 2: Löschen eines Objekts (versionierter Bucket)

Das folgende Beispiel löscht ein Objekt aus einem versionierten Bucket. Das Beispiel löscht eine spezifische Objektversion durch Angabe des Namens des Objektschlüssels und der Versions-ID.

Das Beispiel erledigt Folgendes:

1. Fügt dem Bucket ein Beispielobjekt hinzu. Amazon S3 gibt die die Versions-ID des neu hinzugefügten Objekts zurück. Das Beispiel verwendet diese Versions-ID in der Löschanfrage.
2. Löscht die Objektversion durch Angabe sowohl des Objektschlüsselnamens als auch einer Versions-ID. Wenn keine anderen Versionen dieses Objekts vorhanden sind, löscht Amazon S3 das gesamte Objekt. Andernfalls löscht Amazon S3 die angegebene Version.

 Note

Sie können die Version IDs eines Objekts abrufen, indem Sie eine `ListVersions` Anfrage senden.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.DeleteVersionRequest;
import com.amazonaws.services.s3.model.PutObjectResult;

import java.io.IOException;

public class DeleteObjectVersionEnabledBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Check to ensure that the bucket is versioning-enabled.
            String bucketVersionStatus =
s3Client.getBucketVersioningConfiguration(bucketName).getStatus();
            if (!bucketVersionStatus.equals(BucketVersioningConfiguration.ENABLED))
{
                System.out.printf("Bucket %s is not versioning-enabled.",
bucketName);
            } else {
                // Add an object.
            }
        }
    }
}
```

```
        PutObjectResult putResult = s3Client.putObject(bucketName, keyName,
            "Sample content for deletion example.");
        System.out.printf("Object %s added to bucket %s\n", keyName,
bucketName);

        // Delete the version of the object that we just created.
        System.out.println("Deleting versioned object " + keyName);
        s3Client.deleteVersion(new DeleteVersionRequest(bucketName, keyName,
putResult.getVersionId()));
        System.out.printf("Object %s, version %s deleted\n", keyName,
putResult.getVersionId());
    }
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Das folgende Beispiel veranschaulicht, wie Sie ein Objekt sowohl aus versionierten als auch aus nicht-versionierten Buckets löschen. Weitere Informationen über das S3-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Example Löschen eines Objekts aus einem nicht-versionierten Bucket

Das folgende C#-Beispiel löscht ein Objekt aus einem nicht-versionierten Bucket. In dem Beispiel wird davon ausgegangen, dass die Objekte keine Version habenIDs, sodass Sie keine Version angebenIDs. Sie geben nur den Objektschlüssel ein.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
```

```
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DeleteObjectNonVersionedBucketTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string keyName = "**** object key ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            DeleteObjectNonVersionedBucketAsync().Wait();
        }
        private static async Task DeleteObjectNonVersionedBucketAsync()
        {
            try
            {
                var deleteObjectRequest = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(deleteObjectRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
        }
    }
}
```

```
}
```

Example Löschen eines Objekts aus einem versionierten Bucket

Das folgende C#-Beispiel löscht ein Objekt aus einem versionierten Bucket. Es löscht eine bestimmte Objektversion durch Angabe des Namens des Objektschlüssels und der Versions-ID.

Der Code führt die folgenden Aufgaben durch:

1. Aktiviert das S3-Versioning für einen Bucket, den Sie angeben (wenn S3-Versioning bereits aktiviert ist, hat dies keine Auswirkungen).
2. Fügt dem Bucket ein Beispielobjekt hinzu. In der Antwort gibt Amazon S3 die Versions-ID des neu hinzugefügten Objekts zurück. Das Beispiel verwendet diese Versions-ID in der Löschanfrage.
3. Löscht das Beispielobjekt durch Angabe sowohl des Objektschlüsselnamens als auch einer Versions-ID.

Note

Sie können die Version-IDs eines Objekts durch Senden einer `ListVersions`-Anfrage anfordern.

```
var listResponse = client.ListVersions(new ListVersionsRequest { BucketName  
    = bucketName, Prefix = keyName });
```

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;  
using Amazon.S3;  
using Amazon.S3.Model;  
using System;  
using System.Threading.Tasks;  
  
namespace Amazon.DocSamples.S3  
{  
    class DeleteObjectVersion  
    {  
        private const string bucketName = "*** versioning-enabled bucket name ***";
```

```
private const string keyName = "**** Object Key Name ****";
// Specify your bucket region (an example region is shown).
private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
private static IAmazonS3 client;

public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    CreateAndDeleteObjectVersionAsync().Wait();
}

private static async Task CreateAndDeleteObjectVersionAsync()
{
    try
    {
        // Add a sample object.
        string versionID = await PutAnObject(keyName);

        // Delete the object by specifying an object key and a version ID.
        DeleteObjectRequest request = new DeleteObjectRequest
        {
            BucketName = bucketName,
            Key = keyName,
            VersionId = versionID
        };
        Console.WriteLine("Deleting an object");
        await client.DeleteObjectAsync(request);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
deleting an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
deleting an object", e.Message);
    }
}

static async Task<string> PutAnObject(string objectKey)
{
    PutObjectRequest request = new PutObjectRequest
```

```
        {
            BucketName = bucketName,
            Key = objectKey,
            ContentBody = "This is the content body!"
        };
        PutObjectResponse response = await client.PutObjectAsync(request);
        return response.VersionId;
    }
}
```

PHP

Dieses Beispiel zeigt, wie Sie Klassen aus Version 3 von verwenden AWS SDK for PHP um ein Objekt aus einem Bucket ohne Versionsnummer zu löschen. Weitere Informationen über das Löschen eines Objekts aus einem versionsfähigen Bucket finden Sie unter [Verwenden von REST API](#).

Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Das folgende PHP Beispiel löscht ein Objekt aus einem Bucket. Da dieses Beispiel zeigt, wie Objekte aus nicht versionsgesteuerten Buckets gelöscht werden, gibt es in der Löschanfrage nur den Bucket-Namen und Objektschlüssel (keine Versions-ID) an.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// 1. Delete the object from the bucket.
try
```

```
{
    echo 'Attempting to delete ' . $keyname . '...' . PHP_EOL;

    $result = $s3->deleteObject([
        'Bucket' => $bucket,
        'Key'     => $keyname
    ]);

    if ($result['DeleteMarker'])
    {
        echo $keyname . ' was deleted or does not exist.' . PHP_EOL;
    } else {
        exit('Error: ' . $keyname . ' was not deleted.' . PHP_EOL);
    }
}
catch (S3Exception $e) {
    exit('Error: ' . $e->getAwsErrorMessage() . PHP_EOL);
}

// 2. Check to see if the object was deleted.
try
{
    echo 'Checking to see if ' . $keyname . ' still exists...' . PHP_EOL;

    $result = $s3->getObject([
        'Bucket' => $bucket,
        'Key'     => $keyname
    ]);

    echo 'Error: ' . $keyname . ' still exists.';
}
catch (S3Exception $e) {
    exit($e->getAwsErrorMessage());
}
```

Javascript

```
import { DeleteObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "../libs/s3Client.js" // Helper function that creates Amazon
S3 service client module.

export const bucketParams = { Bucket: "BUCKET_NAME", Key: "KEY" };
```



```
export const run = async () => {
  try {
    const data = await s3Client.send(new DeleteObjectCommand(bucketParams));
    console.log("Success. Object deleted.", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Verwendung der AWS CLI

Um ein Objekt pro Anfrage zu löschen, verwenden Sie den DELETEAPI. Weitere Informationen finden Sie unter [DELETEObjekt](#). Weitere Hinweise zur Verwendung von CLI zum Löschen eines Objekts finden Sie unter [delete-object](#).

Verwenden von REST API

Sie können das AWS SDKsum ein Objekt zu löschen. Wenn Ihre Anwendung dies jedoch erfordert, können Sie REST Anfragen direkt senden. Weitere Informationen finden Sie unter [DELETEObjekt](#) in der Amazon Simple Storage Service API Reference.

Löschen von mehreren Objekten


Für alle Objekte, die Sie in Ihrem S3-Bucket aufbewahren, entstehen Speicherkosten, deshalb sollten Sie Objekte löschen, die Sie nicht mehr brauchen. Wenn Sie beispielsweise Protokolldateien sammeln, sollten Sie sie unbedingt löschen, wenn sie nicht mehr benötigt werden. Sie können einen Lebenszyklusregel definieren, um Objekte wie Protokolldateien automatisch zu löschen. Weitere Informationen finden Sie unter [the section called "Einrichten der Lebenszyklus-Konfiguration"](#).

Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon-S3-Preise](#).

Sie können die Amazon S3 S3-Konsole verwenden, AWS SDKs, oder REST API um mehrere Objekte gleichzeitig aus einem S3-Bucket zu löschen.


Verwenden der S3-Konsole

Befolgen Sie diese Schritte, um mithilfe der Amazon-S3-Konsole mehrere Objekte aus einem Bucket zu löschen.

 Warning

- Das Löschen eines angegebenen Objekts kann nicht rückgängig gemacht werden.
- Diese Aktion löscht alle angegebenen Objekte. Warten Sie beim Löschen von Ordnern, bis die Löschaktion abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte gelöscht werden.
- Wenn Objekte in einem Bucket ohne aktivierte Versionierung gelöscht werden, löscht Amazon S3 die Objekte dauerhaft.
- Beim Löschen von Objekten in einem Bucket mit aktivierter oder angehaltener Bucket-Versionierung erstellt Amazon S3 Löschmarkierungen. Weitere Informationen finden Sie unter [Arbeiten mit Löschmarkierungen](#).

Um Objekte zu löschen, für die die Versionsverwaltung aktiviert oder gesperrt ist

 Note

Wenn die Version IDs für das Objekt in einem Bucket, für das die Versionierung gesperrt wurde NULL, als markiert sind, löscht S3 die Objekte dauerhaft, da keine früheren Versionen existieren. Wenn jedoch eine gültige Versions-ID für die Objekte in einem Bucket aufgeführt ist, für das die Versionierung gesperrt wurde, erstellt S3 Löschmarkierungen für die gelöschten Objekte, wobei die vorherigen Versionen der Objekte beibehalten werden.

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste der Bucket-Namen den Namen des Buckets aus, aus dem Sie die Objekte löschen möchten.
3. Wählen Sie die Objekte aus und klicken Sie dann auf Löschen.
4. Um das Löschen der Objektliste unter Bestimmte Objekte im Bereich Objekte löschen zu bestätigen? Geben Sie in das Textfeld ein **delete**.

Um bestimmte Objektversionen in einem Bucket mit aktivierter Versionierung dauerhaft zu löschen

Warning

Wenn Sie bestimmte Objektversionen in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste der Bucket-Namen den Namen des Buckets aus, aus dem Sie die Objekte löschen möchten.
3. Wählen Sie das zu löschende -Objekt aus.
4. Wählen Sie den Schalter Versionen anzeigen.
5. Wählen Sie die Objektversionen aus und wählen Sie dann Löschen.
6. Um das dauerhafte Löschen der spezifischen Objektversionen zu bestätigen, die unter Bestimmte Objekte im Bereich Objekte löschen? aufgeführt sind Geben Sie im Textfeld Dauerhaft löschen ein. Amazon S3 löscht die spezifischen Objektversionen dauerhaft.

Um die Objekte in einem Amazon S3 S3-Bucket, für die die Versionierung nicht aktiviert ist, dauerhaft zu löschen

Warning

Wenn Sie ein Objekt in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden. Außerdem sind Löschungen für alle Buckets ohne aktivierte Versionierung dauerhaft.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste der Bucket-Namen den Namen des Buckets aus, aus dem Sie die Objekte löschen möchten.
3. Wählen Sie die Objekte aus und klicken Sie dann auf Löschen.
4. Um das dauerhafte Löschen der unter Bestimmte Objekte aufgelisteten Objekte im Abschnitt Objekte löschen? zu bestätigen Geben Sie im Textfeld Dauerhaft löschen ein.

Note

Falls beim Löschen Ihrer Objekte Probleme auftreten, finden Sie weitere Informationen unter [Ich möchte versionierte Objekte dauerhaft löschen](#).

Verwendung der AWS SDKs

Beispiele dafür, wie Sie mehrere Objekte löschen können mit AWS SDKs, siehe [Verwenden Sie DeleteObjects mit einem AWS SDK oder CLI](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Mit dem REST API

Sie können das AWS SDKsum mehrere Objekte mit dem Multi-Object Delete zu löschenAPI. Wenn Ihre Anwendung dies jedoch erfordert, können Sie REST Anfragen direkt senden.

Weitere Informationen finden Sie unter [Mehrere Objekte löschen](#) in der Amazon Simple Storage API Service-Referenz.

Organisieren, Auflisten und Arbeiten mit Ihren Objekten

In Amazon S3 können Sie Präfixe verwenden, um Ihren Speicher zu organisieren. Ein Präfix ist eine logische Gruppierung der Objekte in einem Bucket. Der Präfixwert ist mit einem Verzeichnisnamen vergleichbar, der Ihnen ermöglicht, ähnliche Daten in demselben Verzeichnis eines Buckets zu speichern. Wenn Sie Objekte programmgesteuert hochladen, können Sie Präfixe verwenden, um Ihre Daten zu organisieren.

In der Amazon-S3-Konsole werden Präfixe als Ordner bezeichnet. Sie können alle Ihre Objekte und Ordner in der S3-Konsole anzeigen, indem Sie zu einem Bucket navigieren. Sie können auch Informationen zu jedem Objekt anzeigen, einschließlich Objekteigenschaften.

Weitere Informationen zum Auflisten und Organisieren Ihrer Daten in Amazon S3 finden Sie in den folgenden Themen.

Themen

- [Organisieren von Objekten mit Präfixen](#)
- [Programmgesteuertes Auflisten von Objektschlüsseln](#)

- [Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern](#)
- [Anzeigen einer Objektübersicht in der Amazon-S3-Konsole](#)
- [Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole](#)

Organisieren von Objekten mit Präfixen

Sie können Präfixe verwenden, um die Daten zu organisieren, die Sie in Amazon-S3-Buckets speichern. Ein Präfix ist eine Zeichenfolge am Anfang des Objektschlüsselnamens. Ein Präfix kann eine beliebige Länge haben, abhängig von der maximalen Länge des Objektschlüsselnamens (1.024 Byte). Sie können sich Präfixe als eine Möglichkeit vorstellen, Ihre Daten ähnlich wie Verzeichnisse zu organisieren. Präfixe sind jedoch keine Verzeichnisse.

Die Suche nach Präfix begrenzt die Ergebnisse auf die Schlüssel, die mit dem angegebenen Präfix beginnen. Das Trennzeichen veranlasst Listenoperationen, alle Schlüssel mit gemeinsamem Präfix in einer Zusammenfassungsliste als Ergebnis bereitzustellen.

Zweck der Präfix- und Trennzeichen-Parameter ist es, Ihnen dabei zu helfen, Ihre Schlüssel hierarchisch zu organisieren und dann zu durchsuchen. Dazu wählen Sie zuerst ein Trennzeichen für Ihren Bucket aus, wie beispielsweise einen Schrägstrich (/), das in Ihren Schlüsselnamen voraussichtlich nicht vorkommt. Sie können ein anderes Zeichen als Trennzeichen verwenden. Der Schrägstrich (/) hat keine besonderen Eigenschaften, wird aber sehr häufig als Präfix-Trennzeichen verwendet. Anschließend konstruieren Sie Ihre Schlüsselnamen, indem Sie alle Ebenen der Hierarchie verknüpfen und jede Ebene mit dem Trennzeichen abtrennen.

Wenn Sie beispielsweise Informationen über Städte speichern, könnten Sie diese natürlich nach dem Kontinent, dann nach dem Land und dann nach der Provinz oder dem Staat anordnen. Diese Namen enthalten normalerweise keine Interpunktionszeichen, deshalb könnten Sie den Schrägstrich (/) als Trennzeichen verwenden. Die folgenden Beispiele verwenden einen Schrägstrich (/) als Trennzeichen.

- Europa/Frankreich/Nouvelle-Aquitaine/Bordeaux
- North America/Canada/Quebec/Montreal
- Nordamerika/ /Washington/Bellevue USA
- Nordamerika/ USA /Washington/Seattle

Wenn Sie auf diese Weise Daten für jede Stadt in der Welt gespeichert haben, ist es relativ mühsam, einen flachen Schlüssel-namespace zu verwalten. Durch Verwendung von `Prefix` und `Delimiter`

für die Listenvorgänge können Sie die Hierarchie verwenden, die Sie zum Auflisten Ihrer Daten erstellt haben. Um beispielsweise alle Bundesstaaten in aufzulisten, legen Sie fest. USA
Delimiter='/' Prefix='North America/USA/' Um alle Provinzen in Kanada aufzulisten, für die Sie Daten haben, legen Sie Delimiter='/' und Prefix='North America/Canada/' fest.

Weitere Informationen zu Trennzeichen, Präfixen und verschachtelten Ordnern finden Sie unter [Unterschied zwischen Präfixen und verschachtelten Ordnern](#).

Auflisten von Objekten mit Präfixen und Trennzeichen

Wenn Sie eine Listenanforderung mit einem Trennzeichen ausgeben, können Sie Ihre Hierarchie nur auf einer Ebene durchsuchen und die (möglicherweise Millionen von) Schlüssel, die auf tieferen Ebenen verschachtelt sind, überspringen und zusammenfassen. Nehmen wir zum Beispiel an, Sie haben einen Bucket (*DOC-EXAMPLE-BUCKET*) mit den folgenden Schlüsseln:

sample.jpg

photos/2006/January/sample.jpg

photos/2006/February/sample2.jpg

photos/2006/February/sample3.jpg

photos/2006/February/sample4.jpg

Der Beispiel-Bucket hat nur das sample.jpg-Objekt auf der Root-Ebene. Um nur die Objekte auf Stammebene im Bucket aufzulisten, senden Sie eine GET Anfrage für den Bucket mit dem Schrägstrich (/) als Trennzeichen. In der Antwort gibt Amazon S3 den sample.jpg-Objektschlüssel zurück, weil er das Trennzeichen / nicht enthält. Alle anderen Schlüssel enthalten das Trennzeichen. Amazon S3 gruppiert diese Schlüssel und gibt ein einziges CommonPrefixes-Element mit dem Präfix-Wert photos/ zurück, eine Unterzeichenfolge vom Anfang dieser Schlüssel bis zum Auftreten des angegebenen Trennzeichens.

Example

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>DOC-EXAMPLE-BUCKET</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
```

```
<Delimiter>/</Delimiter>
<IsTruncated>>false</IsTruncated>
<Contents>
  <Key>sample.jpg</Key>
  <LastModified>2011-07-24T19:39:30.000Z</LastModified>
  <ETag>&quot;d1a7fb5eab1c16cb4f7cf341cf188c3d&quot;</ETag>
  <Size>6</Size>
  <Owner>
    <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>displayname</DisplayName>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<CommonPrefixes>
  <Prefix>photos</Prefix>
</CommonPrefixes>
</ListBucketResult>
```

Weitere Hinweise zum programmgesteuerten Auflisten von Objektschlüsseln finden Sie unter [Programmgesteuertes Auflisten von Objektschlüsseln](#).

Programmgesteuertes Auflisten von Objektschlüsseln

In Amazon S3 können Schlüssel nach Präfix aufgelistet werden. Sie können ein allgemeines Präfix für die Namen verwandter Schlüssel wählen und diese Schlüssel mit einem Sonderzeichen markieren, das die Hierarchie begrenzt. Sie können dann die Listenoperation verwenden, um Schlüssel hierarchisch auszuwählen und zu durchsuchen. In ähnlicher Weise werden Dateien in einem Dateisystem in Ordnern gespeichert.

Amazon S3 stellt eine Listenoperation bereit, die Ihnen ermöglicht, die in einem Bucket enthaltenen Schlüssel aufzulisten. Schlüssel werden nach Bucket und Präfix für die Auflistung ausgewählt. Angenommen, wir haben einen Bucket namens "dictionary", der einen Schlüssel für jedes englische Wort enthält. Sie können einen Aufruf ausführen, um alle Schlüssel in dem Bucket aufzulisten, die mit dem Buchstaben "q" beginnen. Listenergebnisse werden immer in binärer Reihenfolge von UTF -8 zurückgegeben.

SOAPSowohl die Operationen als auch die REST Listenoperation geben ein XML Dokument zurück, das die Namen der passenden Schlüssel und Informationen über das Objekt enthält, das durch die einzelnen Schlüssel identifiziert wird.

Note

SOAP Support Over HTTP ist veraltet, aber SOAP immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstützt SOAP. Anstatt zu verwenden SOAP, empfehlen wir, entweder den REST API oder den AWS SDKs.

Gruppen von Schlüsseln, die ein gemeinsames Präfix haben, abgeschlossen mit einem speziellen Trennzeichen, können in einer Auflistung nach diesem gemeinsamen Präfix ausgegeben werden. Auf diese Weise können Anwendungen ihre Schlüssel hierarchisch organisieren und durchsuchen, ähnlich wie bei der Organisation Ihrer Dateien in Verzeichnissen in einem Dateisystem.

Um beispielsweise den dictionary-Bucket zu erweitern, um mehr als nur englische Wörter aufzunehmen, könnten Sie Schlüssel bilden, indem Sie jedem Wort ein Präfix für seine Sprache und ein Trennzeichen voranstellen, wie beispielsweise "French/loGical". Mit Hilfe dieses Namensschemas und der Funktion zur hierarchischen Auflistung könnten Sie eine Liste nur mit französischen Wörtern abrufen. Sie könnten auch auf oberster Ebene eine Liste verfügbarer Sprachen durchsuchen, ohne alle lexikographischen Schlüssel durchlaufen zu müssen. Weitere Informationen über diesen Aspekt der Auflistung finden Sie unter [Organisieren von Objekten mit Präfixen](#).

REST API

Wenn Ihre Bewerbung dies erfordert, können Sie REST Anfragen direkt senden. Sie können eine GET Anfrage senden, um einige oder alle Objekte in einem Bucket zurückzugeben, oder Sie können Auswahlkriterien verwenden, um eine Teilmenge der Objekte in einem Bucket zurückzugeben. Weitere Informationen finden Sie unter [GETBucket \(List Objects\) Version 2](#) in der Amazon Simple Storage Service API Reference.

Effiziente Implementierung von Listen

Die Listenleistung wird von der Gesamtzahl der Schlüssel in Ihrem Bucket nicht wesentlich beeinflusst. Es ist auch nicht von der Anwesenheit oder Abwesenheit der Argumente `prefix`, `marker`, `maxkeys` oder `delimiter` betroffen.

Durchlaufen mehrseitiger Ergebnisse

Buckets können nahezu unbegrenzt viele Schlüssel enthalten, die vollständigen Ergebnisse einer Listenabfrage können deshalb extrem umfangreich sein. Um große Ergebnismengen zu verwalten, API unterstützt Amazon S3 die Paginierung, um sie in mehrere Antworten aufzuteilen. Jede Antwort

für Listenschlüssel gibt eine Seite mit bis zu 1000 Schlüsseln zurück, mit einer Angabe, ob die Antwort gekürzt wurde. Sie senden immer wieder Listenschlüssel-Anforderungen, bis Sie alle Schlüssel erhalten haben. AWS SDK-Wrapper-Bibliotheken bieten dieselbe Paginierung.

Beispiele

Die folgenden Codebeispiele zeigen, wie man es benutzt. `ListObjects`

CLI

AWS CLI

Im folgenden Beispiel wird der `list-objects` Befehl verwendet, um die Namen aller Objekte im angegebenen Bucket anzuzeigen:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

Im Beispiel wird das `--query` Argument verwendet, um die Ausgabe von `list-objects` nach Schlüsselwert und Größe für jedes Objekt zu filtern.

Weitere Informationen zu Objekten finden Sie unter [Working with Amazon S3 Objects](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie [ListObjects](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl ruft die Informationen über alle Elemente im Bucket „test-files“ ab.

```
Get-S3Object -BucketName test-files
```

Beispiel 2: Dieser Befehl ruft die Informationen über das Objekt "sample.txt" aus dem Bucket „test-files“ ab.

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Beispiel 3: Dieser Befehl ruft die Informationen über alle Elemente mit dem Präfix „sample“ aus dem Bucket „test-files“ ab.

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- API-Einheiten finden Sie unter [ListObjects](#) AWS Tools for PowerShell Cmdlet-Referenz.

Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern

In Amazon S3 sind Buckets und Objekte die primären Ressourcen, in denen Objekte in Buckets gespeichert werden. Amazon S3 besitzt eine flache Struktur statt einer Hierarchie, wie Sie sie in der Regel aus Dateisystemen kennen. Um eine möglichst einfache Organisation zu ermöglichen, unterstützt die Amazon-S3-Konsole jedoch das Ordner-Konzept, um Objekte gruppieren zu können. Die Konsole verwendet dazu ein gemeinsames Namenspräfix für die gruppierten Objekte. Mit anderen Worten, die gruppierten Objekte haben Namen, die mit einer bestimmten Zeichenfolge beginnen. Diese gemeinsame Zeichenfolge oder das gemeinsame Präfix ist der Ordnername. Objektnamen werden auch als Schlüsselnamen bezeichnet.

Beispielsweise können Sie einen Ordner namens photos in der Konsole erstellen und ein Objekt namens myphoto.jpg darin speichern. Das Objekt wird dann mit dem Schlüsselnamen photos/myphoto.jpg gespeichert, wobei photos/ das Präfix ist.

Nachfolgend zwei weitere Beispiele:

- Wenn Sie drei Objekte in Ihrem Bucket haben – logs/date1.txt, logs/date2.txt und logs/date3.txt –, zeigt die Konsole einen Ordner namens logs an. Wenn Sie den Ordner in der Konsole öffnen, sehen Sie drei Objekte: date1.txt, date2.txt und date3.txt.
- Wenn Sie ein Objekt namens photos/2017/example.jpg haben, zeigt die Konsole den Ordner photos an, der den Ordner 2017 enthält. Der Ordner 2017 enthält das Objekt example.jpg.

Sie können Ordner innerhalb von Ordnern anlegen, aber keine Buckets innerhalb von Buckets. Sie können Objekte direkt in einen Ordner hochladen und kopieren. Ordner können erstellt, gelöscht und veröffentlicht werden, sie können jedoch nicht umbenannt werden. Objekte können von einem Ordner in einen anderen kopiert werden.

Important

Wenn Sie einen Ordner in Amazon S3 anlegen, erstellt S3 ein 0-Byte-Objekt mit einem Schlüssel, der auf den von Ihnen angegebenen Ordnernamen festgelegt ist. Wenn Sie beispielsweise einen Ordner mit dem Namen photos in Ihrem Bucket erstellen, erstellt die

Amazon-S3-Konsole ein 0-Byte-Objekt mit dem Schlüssel photos/. Die Konsole erstellt dieses Objekt, um das Ordnerkonzept zu unterstützen.

Die Amazon-S3-Konsole behandelt alle Objekte mit einem Schrägstrich (/) als letztes Zeichen im Schlüsselnamen als Ordner (z. B. examplekeyname/). Es ist nicht möglich, ein Objekt über die Amazon-S3-Konsole hochzuladen, dessen Schlüsselname mit einem / endet. Sie können jedoch Objekte, die mit einem abschließenden Namen benannt sind, / mit dem Amazon S3 hochladen, API indem Sie AWS Command Line Interface (AWS CLI), AWS SDKs, oder RESTAPI.

Ein Objekt mit einem angehängten / im Namen wird in der Amazon-S3-Konsole als Ordner angezeigt. Die Amazon-S3-Konsole zeigt für diese Objekte keinen Inhalt und keine Metadaten an. Wenn Sie ein Objekt mit einem angehängten / im Namen kopieren, wird am Zielstandort ein neuer Ordner erstellt, aber die Daten und Metadaten des Objekts werden nicht kopiert.

Themen

- [Erstellen eines Ordners](#)
- [Veröffentlichen von Ordnern](#)
- [Berechnen der Ordnergröße](#)
- [Löschen von Ordnern](#)

Erstellen eines Ordners

In diesem Abschnitt wird beschrieben, wie Sie mit der Amazon-S3-Konsole einen Ordner erstellen.

Important

Wenn Ihre Bucket-Richtlinie das Hochladen von Objekten in diesen Bucket ohne Tags, Metadaten oder Empfänger der Zugriffskontrollliste (ACL) verhindert, können Sie mit dem folgenden Verfahren keinen Ordner erstellen. Laden Sie stattdessen einen leeren Ordner hoch und geben Sie die folgenden Einstellungen in der Upload-Konfiguration an.

Einen Ordner erstellen

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in dem Sie einen Ordner erstellen möchten.
4. Wenn Ihre Bucket-Richtlinie das Hochladen von Objekten ohne Verschlüsselung in diesen Bucket verhindert, müssen Sie Enable (Aktivieren) unter Server-side encryption (Serverseitige Verschlüsselung) wählen.
5. Wählen Sie Create folder.
6. Geben Sie einen Namen für den Ordner ein (z. B. **favorite-pics**). Wählen Sie dann Create folder (Ordner erstellen).

Veröffentlichen von Ordnern

Wir empfehlen, jeden öffentlichen Zugriff auf Ihre Amazon-S3-Ordner und Buckets zu blockieren, sofern Sie nicht speziell einen öffentlichen Ordner oder Bucket benötigen. Wenn Sie einen Ordner öffentlich machen, kann jeder Benutzer im Internet alle im Ordner gruppierten Objekte anzeigen.

In der Amazon-S3-Konsole können Sie einen Ordner öffentlich machen. Außerdem können Sie einen Ordner öffentlich machen, indem Sie eine Bucket-Richtlinie erstellen, die den Datenzugriff durch ein Präfix einschränkt. Weitere Informationen finden Sie unter [Identity and Access Management für Amazon S3](#).

Warning

Nachdem Sie einen Ordner in der Amazon-S3-Konsole öffentlich gemacht haben, kann er nicht mehr auf privat gesetzt werden. Stattdessen müssen Sie Berechtigungen für jedes einzelne Objekt im öffentlichen Ordner festlegen, damit die Objekte keinen öffentlichen Zugriff haben. Weitere Informationen finden Sie unter [Konfiguration ACLs](#).

Themen

- [Berechnen der Ordnergröße](#)
- [Löschen von Ordnern](#)

Berechnen der Ordnergröße

In diesem Abschnitt wird beschrieben, wie Sie mit der Amazon-S3-Konsole die Größe eines Ordners berechnen.

So berechnen Sie die Größe eines Ordners

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in dem Ihr Ordner gespeichert ist.
4. Aktivieren Sie in der Liste Objects (Objekte) das Kontrollkästchen neben dem Namen des Ordners.
5. Wählen Sie Actions (Aktionen) und dann Calculate total size (Gesamtgröße berechnen) aus.

Note

Wenn Sie die Seite verlassen, sind die Ordnerinformationen (einschließlich der Gesamtgröße) danach nicht mehr verfügbar. Sie müssen die Gesamtgröße erneut berechnen, wenn Sie sie erneut sehen möchten.

Important

Wenn Sie die Aktion Calculate total size (Gesamtgröße berechnen) für bestimmte Objekte oder Ordner in Ihrem Bucket verwenden, berechnet Amazon S3 die Gesamtzahl der Objekte und die Gesamtspeichergröße. Unvollständige oder in Bearbeitung befindliche mehrteilige Uploads und vorherige oder nicht aktuelle Versionen werden jedoch nicht auf die Gesamtzahl der Objekte oder die Gesamtgröße angerechnet. Diese Aktion berechnet nur die Gesamtzahl der Objekte und die Gesamtgröße für die aktuelle oder neueste Version jedes Objekts, das im Bucket gespeichert ist.

Wenn es beispielsweise zwei Versionen eines Objekts in Ihrem Bucket gibt, zählt der Speicherrechner in Amazon S3 diese nur als ein Objekt. Daher kann die Gesamtzahl der Objekte, die in der Amazon S3-Konsole berechnet wird, von der in S3 Storage Lens angezeigten Metrik Objektanzahl und von der in der CloudWatch Amazon-Metrik, gemeldeten Anzahl `abweichenNumberof0bjects` abweichen. Ebenso kann die Gesamtspeichergröße auch von der

Metrik „Gesamtspeicher“ in S3 Storage Lens und von der in angezeigten BucketSizeBytes Metrik abweichen CloudWatch.

Löschen von Ordnern

In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole Ordner aus einem S3-Bucket löschen.

Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter [Amazon S3](#).

Ordner aus einem S3-Bucket löschen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, aus dem Sie Ordner löschen möchten.
3. Aktivieren Sie in der Liste Objects (Objekte) das Kontrollkästchen neben den Ordnern und Objekten, die Sie löschen möchten.
4. Wählen Sie Löschen.
5. Stellen Sie auf der Seite Delete objects (Objekte löschen) sicher, dass die Namen der Ordner, die Sie zum Löschen ausgewählt haben, aufgeführt sind.
6. Geben Sie im Feld Delete objects (Objekte löschen) ein **delete**, und wählen Sie Delete objects (Objekte löschen) aus.

Warning

Diese Aktion löscht alle angegebenen Objekte. Warten Sie beim Löschen von Ordnern, bis die Löschaktion abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte gelöscht werden.

Anzeigen einer Objektübersicht in der Amazon-S3-Konsole

Sie können die Amazon-S3-Konsole verwenden, um einen Überblick über ein Objekt zu erhalten. Die Konsole liefert alle wesentlichen Informationen für ein Objekt an einem Ort.

So öffnen Sie die Detailseite für ein Objekt

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus, für das Sie eine Übersicht erhalten möchten.

Die Seite mit Objektdetails wird geöffnet.

4. Um das Objekt herunterzuladen, wählen Sie Object actions (Objektaktionen) und dann Download (Herunterladen). Um den Pfad des Objekts in die Zwischenablage zu kopieren URL, wählen Sie unter Objekt die URL Option.
5. Wenn Ihr Bucket versionsfähig ist, können Sie Versions (Versionen) wählen, um alle Versionen des Objekts anzuzeigen.
 - Um eine Objektversion herunterzuladen, aktivieren Sie das Kontrollkästchen neben der Versions-ID, wählen Sie Actions (Aktionen) und dann Download (Herunterladen).
 - Um eine Objektversion zu löschen, aktivieren Sie das Kontrollkästchen neben der Versions-ID und wählen Sie Delete (Löschen).

Important

Sie können den Löschvorgang für ein Objekt nur rückgängig machen, wenn seine aktuelle Version gelöscht wurde. Es ist nicht möglich, das Löschen einer vorherigen Version eines Objekts rückgängig zu machen, das gelöscht wurde.

Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole

Sie können die Amazon-S3-Konsole verwenden, um die Eigenschaften eines Objekts anzuzeigen, einschließlich Speicherklasse, Verschlüsselungs-Einstellungen, Markierungen und Metadaten.

Die Eigenschaften für ein Objekt anzeigen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.

3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus, dessen Eigenschaften Sie anzeigen wollen.

Die Object overview (Objektübersicht) für Ihr Objekt wird geöffnet. Sie können nach unten scrollen, um die Objekteigenschaften anzuzeigen.

4. Auf der Seite Object overview (Objektübersicht) können Sie die folgenden Eigenschaften für das Objekt konfigurieren.

Note

- Wenn Sie die Eigenschaften Storage Class (Speicherklasse), Encryption (Verschlüsselung) oder Metadata (Metadaten) ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).
- Wenn Sie die Eigenschaften Speicherklasse, Verschlüsselung oder Metadaten für ein Objekt mit benutzerdefinierten Tags ändern, benötigen Sie die `s3:GetObjectTagging` entsprechende Berechtigung. Wenn Sie diese Eigenschaften für ein Objekt ändern, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Berechtigung.

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, werden diese Eigenschaften für das Objekt aktualisiert, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

- a. Storage class (Speicherklasse) – Jedem Objekt in Amazon S3 ist eine Speicherklasse zugeordnet. Welche Speicherklasse Sie wählen, ist davon abhängig, wie häufig Sie auf das Objekt zugreifen. Die Standardspeicherklasse für S3-Objekte ist. STANDARD Sie wählen, welche Speicherlasse verwendet werden soll, wenn Sie ein Objekt hochladen. Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Um die Speicherklasse nach dem Hochladen eines Objekts zu ändern, wählen Sie Storage class (Speicherklasse). Wählen Sie die gewünschte Speicherklasse aus, und wählen Sie dann Save (Speichern).

- b. Serverseitige Verschlüsselungseinstellungen — Sie können serverseitige Verschlüsselung verwenden, um Ihre S3-Objekte zu verschlüsseln. Weitere Informationen finden Sie unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#) oder [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln \(SSE-S3\) angeben](#).
- c. Metadata (Metadaten) – Jedes Objekt in Amazon S3 weist eine Gruppe von Namen-Wert-Paaren auf, die seine Metadaten repräsentieren. Weitere Informationen zum Hinzufügen von Metadaten zu einem S3-Objekt finden Sie unter [Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole](#).
- d. Tags – Sie kategorisieren den Speicher, indem Sie einem S3-Objekt Tags hinzufügen. Weitere Informationen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).
- e. Objektsperre, gesetzliche Aufbewahrung und Aufbewahrung — Sie können verhindern, dass ein Objekt gelöscht wird. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Sie arbeiten mit vorsignierten URLs

Sie können presigned verwendenURLs, um zeitlich begrenzten Zugriff auf Objekte in Amazon S3 zu gewähren, ohne Ihre Bucket-Richtlinie zu aktualisieren. Ein Presigned URL kann in einem Browser eingegeben oder von einem Programm zum Herunterladen eines Objekts verwendet werden. Die vom Vorsignierten verwendeten Anmeldeinformationen URL sind die von AWS Benutzer, der das URL generiert hat.

Sie können Presigned auch verwendenURLs, um es jemandem zu ermöglichen, ein bestimmtes Objekt in Ihren Amazon S3 S3-Bucket hochzuladen. Dies ermöglicht einen Upload, ohne dass eine andere Partei dies tun muss AWS Sicherheitsanmeldedaten oder -berechtigungen. Wenn ein Objekt mit demselben Schlüssel bereits im Bucket vorhanden ist, wie im vorsignierten Bucket angegebenURL, ersetzt Amazon S3 das vorhandene Objekt durch das hochgeladene Objekt.

Sie können das vorsignierte URL Objekt bis zum Ablaufdatum und der Uhrzeit mehrfach verwenden.

Wenn Sie ein vorsigniertes Dokument erstellen URL, müssen Sie Ihre Sicherheitsanmeldedaten und anschließend Folgendes angeben:

- Ein Amazon-S3-Bucket
- Ein Objektschlüssel (Herunterladen: dieses Objekt wird sich in Ihrem Amazon-S3-Bucket befinden; Hochladen: der Name der hochzuladenden Datei)
- Eine HTTP Methode (GET zum Herunterladen von Objekten oder PUT zum Hochladen)
- Ein Ablaufzeitintervall

Derzeit unterstützt Amazon S3 URLs Presigned die Verwendung der folgenden Algorithmen zur Prüfung der Datenintegrität nicht (CRC32, CRC32C, SHA-1, SHA-256), wenn Sie Objekte hochladen. Um die Integrität Ihres Objekts nach dem Hochladen zu überprüfen, können Sie eine MD5 Zusammenfassung des Objekts, wenn Sie es mit einem vorsignierten Objekt hochladen. URL Weitere Informationen über die Objektintegrität finden Sie unter [Überprüfung der Objektintegrität](#).

Themen

- [Wer kann ein vorsigniertes erstellen URL](#)
- [Ablaufzeit für vorsignierte URLs](#)
- [Einschränkung vordefinierter Funktionen URL](#)
- [Objekte mit vorsignierten teilen URLs](#)
- [Objekte mit Vorsignierung hochladen URLs](#)

Wer kann ein vorsigniertes erstellen URL

Jeder, der über gültige Sicherheitsanmeldedaten verfügt, kann eine URL vorsignierte erstellen. Damit jedoch jemand erfolgreich auf ein Objekt zugreifen kann, URL muss das vorsignierte Objekt von einer Person erstellt worden sein, die berechtigt ist, den Vorgang auszuführen, auf dem das Vorsignierte URL basiert.

Die folgenden Typen von Anmeldeinformationen können Sie verwenden, um ein vorsigniertes Objekt zu erstellen: URL

- IAM Instanzprofil — Gültig bis zu 6 Stunden.
- AWS Security Token Service: Bei einer Unterzeichnung mit langfristigen Sicherheitsanmeldedaten bis zu 36 Stunden oder für die Dauer der temporären Anmeldeinformationen gültig, je nachdem, was zuerst endet.

- IAMBenutzer — Gültig bis zu 7 Tage, wenn Sie AWS Signaturversion 4.

Um ein vorsigniertes Dokument zu erstellenURL, das bis zu 7 Tage gültig ist, delegieren Sie zunächst die IAM Benutzeranmeldedaten (den Zugriffsschlüssel und den geheimen Schlüssel) an die Methode, mit der Sie das vorsignierte Dokument erstellen. URL

Note

Wenn Sie eine vorsignierte Datei URL mit temporären Anmeldeinformationen erstellt haben, URL läuft diese ab, wenn die Anmeldeinformationen ablaufen. Im Allgemeinen URL läuft ein vorsigniertes ab, wenn die Anmeldeinformationen, mit denen Sie es erstellt haben, gesperrt, gelöscht oder deaktiviert werden. Dies gilt auch dann, wenn das mit einer URL späteren Ablaufzeit erstellt wurde. Informationen zur Gültigkeitsdauer temporärer Sicherheitsanmeldedaten finden Sie unter Vergleichen [AWS STS APIOperationen](#) im IAMBenutzerhandbuch.

Ablaufzeit für vorsignierte URLs

Ein vorsigniertes URL Dokument bleibt für den Zeitraum gültig, der bei der URL Generierung angegeben wurde. Wenn Sie ein URL mit der Amazon S3 S3-Konsole vorsigniertes erstellen, kann die Ablaufzeit zwischen 1 Minute und 12 Stunden festgelegt werden. Wenn Sie das AWS CLI or AWS SDKs, die Ablaufzeit kann auf bis zu 7 Tage festgelegt werden.

Wenn Sie ein vorsigniertes URL mit einem temporären Token erstellt haben, URL läuft das ab, wenn das Token abläuft. Im Allgemeinen URL läuft ein vorsigniertes ab, wenn die Anmeldeinformationen, mit denen Sie es erstellt haben, gesperrt, gelöscht oder deaktiviert werden. Dies gilt auch dann, wenn das mit einer URL späteren Ablaufzeit erstellt wurde. Weitere Informationen darüber, wie sich die von Ihnen verwendeten Anmeldeinformationen auf die Ablaufzeit auswirken, finden Sie unter [Wer kann ein vorsigniertes erstellen URL](#).

Amazon S3 überprüft das Ablaufdatum und die Uhrzeit eines Signierten zum URL Zeitpunkt der HTTP Anfrage. Beginnt ein Client beispielsweise mit dem Herunterladen einer großen Datei unmittelbar vor der Ablaufzeit, wird der Download auch dann fortgesetzt, wenn die Ablaufzeit während des Downloads verstreicht. Wenn die Verbindung jedoch unterbrochen wird und der Client versucht, den Download nach Ablauf der Zeit erneut zu starten, schlägt der Download fehl.

Einschränkung vordefinierter Funktionen URL

Die Funktionen einer vorsignierten Datei URL werden durch die Berechtigungen des Benutzers begrenzt, der sie erstellt hat. Im Wesentlichen URLs handelt es sich bei vorsignierten Token um Inhaber-Token, die denjenigen Zugriff gewähren, die sie besitzen. Daher empfehlen wir Ihnen, sie angemessen zu schützen. Im Folgenden finden Sie einige Methoden, mit denen Sie die Verwendung Ihrer vorsignierten Daten einschränken können. URLs

AWS Signatur Version 4 (SigV4)

Um ein bestimmtes Verhalten bei der Authentifizierung vorsignierter URL Anfragen durchzusetzen, verwenden Sie AWS In Signature Version 4 (Sigv4) können Sie Bedingungsschlüssel in Bucket-Richtlinien und Access-Point-Richtlinien verwenden. Die folgende Bucket-Richtlinie verwendet die `s3:signatureAge` Bedingung beispielsweise, um jede vorsignierte Amazon S3 URL S3-Anfrage für Objekte im `amzn-s3-demo-bucket1` Bucket abzulehnen, wenn die Signatur älter als 10 Minuten ist. Um dieses Beispiel zu verwenden, ersetzen Sie *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Deny a presigned URL request if the signature is more than 10 min
old",
      "Effect": "Deny",
      "Principal": {"AWS": "*"},
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": 600000
        }
      }
    }
  ]
}
```

Weitere Informationen zu Richtlinienschlüsseln finden Sie im Zusammenhang AWS Signaturversion 4 finden Sie unter [AWS Authentifizierung mit Signature Version 4](#) in der Amazon Simple Storage API Service-Referenz.

Beschränkung der Netzwege

Wenn Sie die Nutzung des vorsignierten URLs und des gesamten Amazon S3 S3-Zugriffs auf bestimmte Netzwerkpfade einschränken möchten, können Sie schreiben AWS Identity and Access Management (IAM) Richtlinien. Sie können diese Richtlinien für den IAM Principal, der den Anruf tätigt, für den Amazon S3 S3-Bucket oder für beide festlegen.

Eine Netzwerkpfadbeschränkung für den IAM Prinzipal erfordert, dass der Benutzer mit diesen Anmeldeinformationen Anfragen vom angegebenen Netzwerk aus stellt. Eine Einschränkung des Buckets oder des Zugriffspunkts erfordert, dass alle Anfragen an diese Ressource aus dem angegebenen Netz stammen. Diese Einschränkungen gelten auch außerhalb des URL vordefinierten Szenarios.

Welchen IAM globalen Bedingungsschlüssel Sie verwenden, hängt vom Typ des Endpunkts ab. Wenn Sie den öffentlichen Endpunkt für Amazon S3 verwenden, verwenden Sie `aws:SourceIp`. Wenn Sie einen Virtual Private Cloud (VPC) -Endpunkt für Amazon S3 verwenden, verwenden Sie `aws:SourceVpc` oder `aws:SourceVpce`.

Gemäß der folgenden IAM Grundsatzerklärung muss der Principal darauf zugreifen AWS nur aus dem angegebenen Netzwerkbereich. Mit dieser Richtlinie müssen alle Zugriffe von diesem Bereich ausgehen. Dies gilt auch für den Fall, dass jemand ein URL für Amazon S3 vorsigniertes Gerät verwendet. Um dieses Beispiel zu verwenden, ersetzen Sie *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
{
  "Sid": "NetworkRestrictionForIAMPrincipal",
  "Effect": "Deny",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
    "BoolIfExists": {"aws:ViaAWSService": "false"}
  }
}
```

Objekte mit vorsignierten teilen URLs

Standardmäßig sind alle Amazon-S3-Objekte privat und nur der Objekteigentümer hat die Berechtigung, darauf zuzugreifen. Der Objekteigentümer kann jedoch Objekte mit anderen

teilen, indem er ein URL vorsigniertes Objekt erstellt. Ein vorsignierter URL Benutzer verwendet Sicherheitsanmeldedaten, um zeitlich begrenzte Berechtigungen zum Herunterladen von Objekten zu gewähren. Das URL kann in einem Browser eingegeben oder von einem Programm zum Herunterladen des Objekts verwendet werden. Die von den Vorsignierten verwendeten Anmeldeinformationen URL sind die von AWS Benutzer, der das URL generiert hat.

Allgemeine Informationen zu Presigned finden Sie URLs unter [Sie arbeiten mit vorsignierten URLs](#).

Mit der Amazon S3 S3-Konsole können Sie ein Objekt erstellen, das URL für die gemeinsame Nutzung vorsigniert ist, ohne Code schreiben zu müssen. AWS Explorer für Visual Studio (Windows) oder AWS Toolkit for Visual Studio Code. Sie können eine vorsignierte Datei auch URL programmgesteuert generieren, indem Sie den AWS Command Line Interface (AWS CLI) oder AWS SDKs.

Verwenden der S3-Konsole

Sie können die Amazon S3 S3-Konsole verwenden, um ein Objekt zu generieren, das URL für die gemeinsame Nutzung vorsigniert ist, indem Sie die folgenden Schritte ausführen. Wenn Sie die Konsole verwenden, beträgt die maximale Ablaufzeit für ein vorsigniertes URL Objekt 12 Stunden ab dem Zeitpunkt der Erstellung.

Um mit der Amazon S3 S3-Konsole ein URL vorsigniertes zu generieren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Bucket-Liste den Namen des Buckets aus, der das Objekt enthält, für das Sie eine URL Vorsignierung wünschen.
4. Wählen Sie in der Objektliste das Objekt aus, für das Sie ein URL vorsigniertes Objekt erstellen möchten.
5. Wählen Sie im Menü „Objektaktionen“ die Option „Mit einem URL vorsignierten Objekt teilen“.
6. Geben Sie an, wie lange das Vorsignierte gültig sein URL soll.
7. Wählen Sie „URLVorsigniertes erstellen“.
8. Wenn eine Bestätigung erscheint, wird diese automatisch in Ihre Zwischenablage kopiert. URL Sie werden eine Schaltfläche sehen, mit der Sie das Vorsignierte kopieren könnenURL, falls Sie es erneut kopieren müssen.

Verwendung der AWS CLI

Das folgende Beispiel AWS CLI Der Befehl generiert ein URL für die gemeinsame Nutzung eines Objekts vorsigniertes Objekt aus einem Amazon S3 S3-Bucket. Wenn Sie das verwenden AWS CLI, die maximale Ablaufzeit für ein vorsigniertes Dokument URL beträgt 7 Tage ab dem Zeitpunkt der Erstellung. Um dieses Beispiel zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3 presign s3://amzn-s3-demo-bucket1/mydoc.txt --expires-in 604800
```

Note

Für alle AWS-Regionen Nach dem 20. März 2019 gestartet, müssen Sie das endpoint-url und AWS-Region zusammen mit der Anfrage angeben. Eine Liste aller Amazon S3 S3-Regionen und -Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der AWS Allgemeine Referenz.

```
aws s3 presign s3://amzn-s3-demo-bucket1/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

Weitere Informationen finden Sie unter [presign](#) in der AWS CLI Befehlsreferenz.

Verwendung der AWS SDKs


Beispiele für die Verwendung von AWS SDKs Informationen zum Generieren eines URL für die gemeinsame Nutzung eines Objekts vorsignierten Objekts finden [Sie unter Erstellen eines URL für Amazon S3 vorsignierten Objekts mithilfe eines AWS SDK](#).

Wenn Sie das verwenden AWS SDKsum eine vorsignierte Datei zu generierenURL, beträgt die maximale Ablaufzeit 7 Tage ab dem Zeitpunkt der Erstellung.

Note


Für alle AWS-Regionen Nach dem 20. März 2019 gestartet, müssen Sie das endpoint-url und AWS-Region zusammen mit der Anfrage angeben. Eine Liste aller Amazon S3 S3-

Regionen und -Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der AWS Allgemeine Referenz.

 Note

Bei der Verwendung des AWS SDKs, das Tagging-Attribut muss ein Header und kein Abfrageparameter sein. Alle anderen Attribute können als Parameter für das URL Vorsignierte übergeben werden.

Verwendung der AWS Toolkit for Visual Studio (Windows)

 Note

Zu diesem Zeitpunkt ist der AWS Toolkit for Visual Studio unterstützt Visual Studio für Mac nicht.

1. Installieren Sie das AWS Toolkit for Visual Studio mithilfe der folgenden Anweisungen [installieren und einrichten Sie das Toolkit for Visual Studio](#) in der AWS Toolkit for Visual Studio Benutzerleitfaden.
2. Connect to (Verbinden mit) AWS mithilfe der folgenden Schritte, [Herstellen einer Verbindung zu AWS](#) in der AWS Toolkit for Visual Studio Benutzerleitfaden.
3. Auf der linken Seite beschriftet AWS Explorer, doppelklicken Sie auf den Bucket, der Ihr Objekt enthält.
4. Klicken Sie mit der rechten Maustaste auf das Objekt, für das ein vorsigniertes Objekt URL generiert werden soll, und wählen Sie Vorsigniert erstellen... URL .
5. Geben Sie im Popup-Fenster das Ablaufdatum und die Uhrzeit für Ihr vorsigniertes URL Produkt ein.
6. Der Objektschlüssel sollte auf der Grundlage des ausgewählten Objekts vorab ausgefüllt werden.
7. Geben Sie GET an, dass dieser vorsignierte Wert für das Herunterladen eines Objekts verwendet URL werden soll.
8. Wählen Sie die Schaltfläche Generate (Generieren) aus.
9. Um das in die Zwischenablage URL zu kopieren, wählen Sie Kopieren.

10. Um das generierte, vorsignierte Objekt zu verwenden, fügen Sie es in die URL in einen beliebigen Browser ein.

Die Verwendung von AWS Toolkit for Visual Studio Code

Wenn Sie Visual Studio Code verwenden, können Sie einen Code generieren, der vorsigniert ist, um ein Objekt gemeinsam mit einer URL zu nutzen, ohne Code schreiben zu müssen, indem Sie AWS Toolkit for Visual Studio Code verwenden. Allgemeine Informationen finden Sie unter [AWS Toolkit for Visual Studio Code](#) in der AWS Toolkit for Visual Studio Code Benutzerleitfaden.

Für Anweisungen zur Installation des AWS Toolkit for Visual Studio Code, siehe [Installation des AWS Toolkit for Visual Studio Code](#) in der AWS Toolkit for Visual Studio Code Benutzerleitfaden.

1. Connect to (Verbinden mit) AWS mithilfe der folgenden Schritte, [Herstellen einer Verbindung zu AWS Toolkit for Visual Studio Code](#) in der AWS Toolkit for Visual Studio Code Benutzerleitfaden.
2. Wählen Sie das AWS Logo auf der linken Seite in Visual Studio Code.
3. Wählen Sie EXPLORER unter S3 aus.
4. Wählen Sie einen Bucket und eine Datei und öffnen Sie das Kontextmenü (rechte Maustaste).
5. Wählen Sie Generate Presigned URL aus und legen Sie dann die Ablaufzeit (in Minuten) fest.
6. Drücken Sie die Eingabetaste, und das vorsignierte URL wird in Ihre Zwischenablage kopiert.

Objekte mit Vorsignierung hochladen URLs

Sie können presigned verwenden, um jemandem URLs zu ermöglichen, ein Objekt in Ihren Amazon S3 S3-Bucket hochzuladen. Wenn Sie ein vorsigniertes Objekt verwenden, können Sie es hochladen, ohne dass eine andere Partei dies tun muss AWS Sicherheitsanmeldedaten oder Berechtigungen. Ein vorsignierter URL Code ist durch die Berechtigungen des Benutzers begrenzt, der ihn erstellt hat. Das heißt, wenn Sie ein für das Hochladen eines Objekts vorsigniertes URL Dokument erhalten, können Sie ein Objekt nur hochladen, wenn der Ersteller des Objekts URL über die erforderlichen Berechtigungen zum Hochladen dieses Objekts verfügt.

Wenn jemand das verwendet, um ein Objekt hochzuladen, erstellt Amazon S3 das Objekt im angegebenen Bucket. Wenn ein Objekt mit demselben Schlüssel, der im vorsignierten Objekt angegeben ist, URL bereits im Bucket vorhanden ist, ersetzt Amazon S3 das vorhandene Objekt durch das hochgeladene Objekt. Nach dem Upload gehört das Objekt dem Bucket-Besitzer.

Allgemeine Informationen zu Presigned URLs finden Sie unter [Sie arbeiten mit vorsignierten URLs](#)

Sie können ein vorsigniertes Objekt URL für das Hochladen eines Objekts erstellen, ohne Code schreiben zu müssen, indem Sie AWS Explorer für Visual Studio. Sie können eine vorsignierte Datei auch URL programmgesteuert generieren, indem Sie AWS SDKs.

Verwendung der AWS Toolkit for Visual Studio (Windows)

Note

Zu diesem Zeitpunkt ist der AWS Toolkit for Visual Studio unterstützt Visual Studio für Mac nicht.

1. Installieren Sie das AWS Toolkit for Visual Studio mithilfe der folgenden Anweisungen [installieren und einrichten Sie das Toolkit for Visual Studio](#) in der AWS Toolkit for Visual Studio Benutzerleitfaden.
2. Connect to (Verbinden mit) AWS mithilfe der folgenden Schritte, [Herstellen einer Verbindung zu AWS](#) in der AWS Toolkit for Visual Studio Benutzerleitfaden.
3. Auf der linken Seite beschriftet AWS Klicken Sie im Explorer mit der rechten Maustaste auf den Bucket, in den Sie ein Objekt hochladen möchten.
4. Wählen Sie Vorsigniert URL erstellen... .
5. Geben Sie im Popup-Fenster das Ablaufdatum und die Uhrzeit für Ihr vorsigniertes URL Produkt ein.
6. Geben Sie für Object Key den Namen der hochzuladenden Datei ein. Die Datei, die Sie hochladen, muss genau diesem Namen entsprechen. Wenn ein Objekt mit demselben Objektschlüssel bereits im Bucket vorhanden ist, ersetzt Amazon S3 das vorhandene Objekt durch das neu hochgeladene Objekt.
7. Geben Sie PUT an, dass dieses vorsignierte URL Objekt für das Hochladen eines Objekts verwendet wird.
8. Wählen Sie die Schaltfläche Generate (Generieren) aus.
9. Um das in die Zwischenablage URL zu kopieren, wählen Sie Kopieren.
10. Um dies zu verwenden URL, können Sie eine PUT Anfrage mit dem `curl` Befehl senden. Geben Sie den vollständigen Pfad zu Ihrer Datei und die vorsignierte Datei URL an.

```
curl -X PUT -T "/path/to/file" "presigned URL"
```

Verwendung der AWS SDKs

Beispiele für die Verwendung von AWS SDKs Informationen zum Generieren einer URL für das Hochladen eines Objekts vorsignierten Datei finden Sie unter [Erstellen einer URL für Amazon S3 vorsignierten Datei mithilfe eines AWS SDK](#).

Wenn Sie das verwenden AWS SDKsum eine vorsignierte Datei zu generierenURL, beträgt die maximale Ablaufzeit 7 Tage ab dem Zeitpunkt der Erstellung.

Note

Für alle AWS-Regionen Nach dem 20. März 2019 gestartet, müssen Sie das `endpoint-url` und `AWS-Region` zusammen mit der Anfrage angeben. Eine Liste aller Amazon S3 S3-Regionen und -Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der AWS Allgemeine Referenz.

Transformieren von Objekten mit S3 Object Lambda

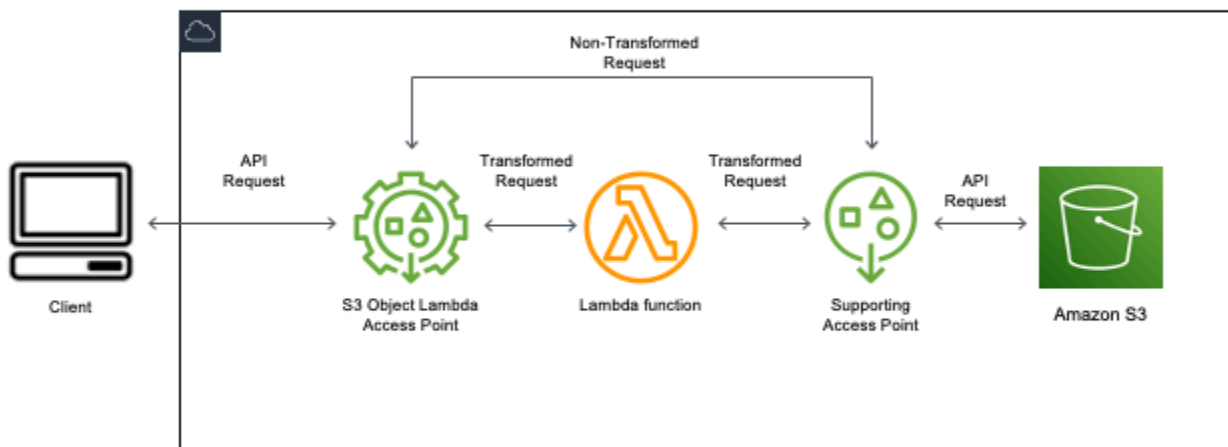
Mit S3 Object Lambda können Sie GET-, LIST- und HEAD-Anforderungen von Amazon S3 eigenen Code hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Sie können mit benutzerdefiniertem Code die von S3-GET-Anforderungen zurückgegebenen Daten ändern, um Zeilen zu filtern, Bilder dynamisch in der Größe zu ändern und mit Wassermarken zu versehen, vertrauliche Daten zu redigieren und mehr. Sie können S3 Object Lambda auch verwenden, um die Ausgabe von S3-LIST-Anforderungen zu ändern und eine benutzerdefinierte Ansicht aller Objekte in einem Bucket zu erstellen, und von S3-HEAD-Anforderungen zum Ändern von Objektmetadaten wie Objektname und -größe. Sie können S3 Object Lambda als Quelle für Ihre CloudFront Amazon-Distribution verwenden, um Daten auf Endbenutzer zuzuschneiden, z. B. die automatische Größenänderung von Bildern, die Transcodierung älterer Formate (wie von JPEG zu WebP) oder das Entfernen von Metadaten. Weitere Informationen finden Sie im AWS Blogbeitrag [Verwenden Sie Amazon S3 Object Lambda mit Amazon CloudFront](#). Bereitgestellt von AWS Lambda-Funktionen, Ihr Code läuft auf einer Infrastruktur, die vollständig verwaltet wird von AWS. Die Verwendung von S3 Object Lambda reduziert die Notwendigkeit, abgeleitete Kopien Ihrer Daten zu erstellen und zu speichern oder Proxys auszuführen, ohne dass Sie Ihre Anwendungen ändern müssen.

So funktioniert S3 Object Lambda

S3-Objekt, das Lambda verwendet AWS Lambda Funktionen zur automatischen Verarbeitung der Ausgabe von Standard-S3- GETLIST, oder HEAD -Anfragen. AWS Lambda ist ein serverloser Rechendienst, der kundendefinierten Code ausführt, ohne dass die Verwaltung der zugrunde liegenden Rechenressourcen erforderlich ist. Sie können Ihre eigenen benutzerdefinierten Lambda-Funktionen erstellen und ausführen und die Datentransformation an Ihren spezifischen Anwendungsfall anpassen.

Nachdem Sie eine Lambda-Funktion konfiguriert haben, fügen Sie sie an einen Serviceendpunkt von S3 Object Lambda an, der als Object Lambda Access Point bezeichnet wird. Der Object Lambda Access Point verwendet für den Zugriff auf Amazon S3 einen standardmäßigen S3-Zugriffspunkt, der als unterstützender Zugriffspunkt bezeichnet wird.

Wenn Sie eine Anforderung an Ihren Object Lambda Access Point senden, ruft Amazon S3 Ihre Lambda-Funktion automatisch auf. Dann geben alle Daten, die mit der S3-Anforderung GET, LIST oder HEAD über den Object Lambda Access Point abgerufen werden, ein transformiertes Ergebnis an die Anwendung zurück. Alle anderen Anforderungen werden wie gewohnt verarbeitet, wie im folgenden Diagramm dargestellt.



Die Themen in diesem Abschnitt beschreiben, wie Sie mit S3 Object Lambda arbeiten.

Themen

- [Erstellen von Objekt-Lambda-Zugriffspunkten](#)
- [Verwenden von Amazon S3 Objekt-Lambda-Zugriffspunkten](#)
- [Sicherheitsüberlegungen für S3 Object Lambda-Zugriffspunkte](#)
- [Schreiben von Lambda-Funktionen für S3 Object Lambda Access Points](#)
- [Die Verwendung von AWS gebaute Lambda-Funktionen](#)
- [Bewährte Methoden und Richtlinien für S3 Object Lambda](#)
- [Tutorials zu S3 Object Lambda](#)
- [Debuggen von S3 Object Lambda](#)

Erstellen von Objekt-Lambda-Zugriffspunkten

Ein Object Lambda Access Point ist mit genau einem Standard-Zugriffspunkt und folglich mit einem Amazon-S3-Bucket verknüpft. Zum Erstellen eines Object Lambda Access Point benötigen Sie folgende Ressourcen:

- Ein Amazon-S3-Bucket Informationen zum Erstellen von Buckets finden Sie unter [the section called “Erstellen eines Buckets”](#).
- Ein standardmäßiger S3-Zugriffspunkt Wenn Sie mit Object Lambda Access Point arbeiten, wird dieser Standardzugriffspunkt als unterstützender Zugriffspunkt bezeichnet. Informationen zum Erstellen von Standardzugriffspunkten finden Sie unter [the section called “Erstellen von Zugriffspunkten”](#).
- Ein AWS Lambda Funktion. Sie können entweder Ihre eigene Lambda-Funktion erstellen oder eine vordefinierte Funktion verwenden. Weitere Informationen zum Erstellen von Lambda-Funktionen finden Sie unter [the section called “Schreiben von Lambda-Funktionen”](#). Weitere Hinweise zu vorgefertigten Funktionen finden Sie unter [Die Verwendung von AWS gebaute Lambda-Funktionen](#).
- (Fakultativ) Ein AWS Identity and Access Management (IAM) Richtlinie. Amazon S3 Access Points unterstützen IAM Ressourcenrichtlinien, mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Weitere Informationen zum Erstellen dieser Richtlinien finden Sie unter [the section called “Richtlinien konfigurieren IAM”](#).

In den folgenden Abschnitten wird beschrieben, wie Sie einen Object Lambda Access Point mit Folgendem erstellen:

- Das Tool AWS Management Console
- Das Tool AWS Command Line Interface (AWS CLI)
- Importieren in S3; AWS CloudFormation Vorlage
- Das Tool AWS Cloud Development Kit (AWS CDK)

Informationen zum Erstellen eines Object Lambda Access Points mithilfe von finden Sie [CreateAccessPointForObjectLambda](#) in der REST API Amazon Simple Storage Service API Reference.

Erstellen eines Object Lambda Access Point

Erstellen Sie Ihren Object Lambda Access Point mit einem der folgenden Verfahren.

Verwenden der S3-Konsole

So erstellen Sie einen Object Lambda Access Point mit der Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste den Namen der aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
4. Wählen Sie auf der Seite Object Lambda access points (Objekt-Lambda-Zugriffspunkte) die Option Create Object Lambda access point (Objekt-Lambda-Zugriffspunkt erstellen).
5. Geben Sie unter Object Lambda access point name (Name des Objekt-Lambda-Zugriffspunkts) den Namen ein, den Sie für den Zugriffspunkt verwenden möchten.


Wie bei Standard-Zugriffspunkten gibt es Regeln für die Benennung von Object Lambda Access Points. Weitere Informationen finden Sie unter [Regeln zur Benennung von Amazon S3-Zugriffspunkten](#).

6. Geben Sie für Supporting Access Point (Unterstützender Access Point) den Standard-Zugriffspunkt ein, den Sie verwenden möchten, oder navigieren Sie zu diesem. Der Access Point muss sich in derselben befinden AWS-Region wie die Objekte, die Sie transformieren möchten. Informationen zum Erstellen von Standardzugriffspunkten finden Sie unter [the section called "Erstellen von Zugriffspunkten"](#).
7. Unter Transformationskonfiguration können Sie eine Funktion hinzufügen, die Ihre Daten für den Object Lambda Access Point transformiert. Führen Sie eine der folgenden Aktionen aus:

- Wenn Sie bereits eine haben AWS Lambda Funktion in Ihrem Konto können Sie sie unter Lambda-Funktion aufrufen auswählen. Hier können Sie den Amazon-Ressourcennamen (ARN) einer Lambda-Funktion in Ihrem AWS-Konto oder wählen Sie eine Lambda-Funktion aus dem Drop-down-Menü.
- Wenn Sie eine verwenden möchten AWS eingebaute Funktion wählen Sie den Funktionsnamen unter AWS erstellte Funktion und wählen Sie Lambda-Funktion erstellen. Dadurch gelangen Sie zur Lambda-Konsole, wo Sie eine integrierte Funktion in Ihrem System bereitstellen können AWS-Konto. Weitere Hinweise zu integrierten Funktionen finden Sie unter [Die Verwendung von AWS gebaute Lambda-Funktionen](#).

Wählen Sie unter S3 eine oder mehrere API Operationen aus APIs, die aufgerufen werden sollen. Für jede API Auswahl müssen Sie eine Lambda-Funktion angeben, die aufgerufen werden soll.

8. (Optional) Fügen Sie unter Payload JSON Text hinzu, den Sie Ihrer Lambda-Funktion als Eingabe zur Verfügung stellen möchten. Sie können Nutzlasten mit verschiedenen Parametern für verschiedene Object Lambda Access Points konfigurieren, die dieselbe Lambda-Funktion aufrufen, wodurch die Flexibilität Ihrer Lambda-Funktion erhöht wird.

 Important

Wenn Sie Object Lambda Access Points verwenden, stellen Sie sicher, dass die Nutzlast keine vertraulichen Informationen enthält.

9. (Optional) Für Range and part number (Bereichs- und Teilenummer) müssen Sie diese Option aktivieren, wenn Sie GET- und HEAD-Anforderungen mit Bereichs- und Teilenummern-Headern bearbeiten möchten. Durch Aktivieren dieser Option wird bestätigt, dass Ihre Lambda-Funktion diese Anforderungen erkennen und bearbeiten kann. Weitere Informationen zu Bereichsüberschriften und Teilenummern finden Sie unter [Arbeitet mit Range and partNumber Header](#).
10. (Optional) Wählen Sie für Anforderungsmetriken die Option Aktivieren oder Deaktivieren aus, um Ihrem Object Lambda Access Point Amazon-S3-Überwachung hinzuzufügen. Anforderungsmetriken werden zum CloudWatch Standardtarif von Amazon abgerechnet.
11. (Optional) Legen Sie unter Object Lambda Access Point policy (Objekt-Lambda-Access-Point-Richtlinie) eine Ressourcenrichtlinie fest. Ressourcenrichtlinien erteilen Berechtigungen für den angegebenen Object Lambda Access Point und können die Verwendung des Zugriffspunkts nach Ressource, Benutzer oder anderen Bedingungen steuern. Weitere Hinweise zu

Ressourcenrichtlinien für Objekt-Lambda-Zugriffspunkten finden Sie unter [Konfiguration von IAM Richtlinien für Object Lambda Access Points](#).

- Wählen Sie unter Block Public Access settings for this Object Lambda Access Point (Einstellungen für die Blockierung des öffentlichen Zugriffs für diesen Objekt-Lambda-Zugriffspunkt) die Einstellungen für die Blockierung des öffentlichen Zugriffs aus, die Sie anwenden möchten. Alle Einstellungen für die Blockierung des öffentlichen Zugriffs sind standardmäßig für neue Objekt-Lambda-Zugriffspunkte aktiviert. Wir empfehlen Ihnen, die Standardeinstellungen aktiviert zu lassen. Amazon S3 unterstützt derzeit nicht das Ändern der Einstellungen für die Blockierung des öffentlichen Zugriffs eines Objekt-Lambda-Zugriffspunkts, nachdem der Objekt-Lambda-Zugriffspunkt erstellt wurde.

Weitere Informationen über die Blockierung des öffentlichen Zugriffs in Amazon S3 finden Sie unter [Verwalten des öffentlichen Zugriffs auf Zugriffspunkte](#).

- Wählen Sie Create Object Lambda Access Point (Objekt-Lambda-Access-Point erstellen) aus.

Verwendung der AWS CLI

So erstellen Sie einen Object Lambda Access Point mit einem AWS CloudFormation Vorlage

Note

Wenn Sie die folgenden Befehle verwenden, ersetzen Sie *user input placeholders* durch eigene Informationen.

- Laden Sie das herunter AWS Lambda Funktionsbereitstellungspaket `s3objectlambda_deployment_package.zip` in der [Standardkonfiguration von S3 Object Lambda](#).
- Führen Sie den folgenden `put-object`-Befehl aus, um das Paket in einen Amazon-S3-Bucket hochzuladen.

```
aws s3api put-object --bucket Amazon S3 bucket name --key  
s3objectlambda_deployment_package.zip --body release/  
s3objectlambda_deployment_package.zip
```

- Laden Sie das herunter AWS CloudFormation Vorlage `s3objectlambda_defaultconfig.yaml` in der [Standardkonfiguration von S3 Object Lambda](#).

4. Führen Sie den folgenden `deploy` Befehl aus, um die Vorlage auf Ihrem Computer bereitzustellen AWS-Konto.

```
aws cloudformation deploy --template-file s3objectlambda_defaultconfig.yaml \
  --stack-name AWS CloudFormation stack name \
  --parameter-overrides ObjectLambdaAccessPointName=Object Lambda Access Point name \
  SupportingAccessPointName=Amazon S3 access point S3BucketName=Amazon S3 bucket \
  LambdaFunctionS3BucketName=Amazon S3 bucket containing your Lambda package \
  LambdaFunctionS3Key=Lambda object key LambdaFunctionS3ObjectVersion=Lambda object version \
  LambdaFunctionRuntime=Lambda function runtime --capabilities capability_IAM
```

Sie können das konfigurieren AWS CloudFormation Vorlage zum Aufrufen von Lambda für GETHEAD, und LIST API -Operationen. Weitere Informationen zum Ändern der Standardkonfiguration der Vorlage finden Sie unter [the section called “Automatisieren Sie die Einrichtung von S3 Object Lambda mit AWS CloudFormation”](#).

So erstellen Sie einen Object Lambda Access Point mit dem AWS CLI

Note

Wenn Sie die folgenden Befehle verwenden, ersetzen Sie *user input placeholders* durch eigene Informationen.

Im folgenden Beispiel wird ein Object Lambda Access Point mit dem Namen erstellt *my-object-lambda-ap* für den Bucket *amzn-s3-demo-bucket1* auf dem Konto *111122223333*. In diesem Beispiel wird davon ausgegangen, dass ein Standardzugriffspunkt mit dem Namen *Beispiel-AP* wurde bereits erstellt. Informationen zum Erstellen eines Standardzugriffspunkts finden Sie unter [the section called “Erstellen von Zugriffspunkten”](#).

Dieses Beispiel verwendet die AWS vorgefertigte Funktion `decompress`. Weitere Hinweise zu vorgefertigten Funktionen finden Sie unter [the section called “Die Verwendung von AWS gebaute Funktionen”](#).

1. Erstellen Sie einen Bucket. In diesem Beispiel verwenden wir *amzn-s3-demo-bucket1*. Hinweise zum Erstellen von Buckets finden Sie unter [the section called “Erstellen eines Buckets”](#).

2. Erstellen Sie einen Standardzugriffspunkt und hängen Sie ihn an Ihren Bucket an.
In diesem Beispiel verwenden wir *Beispiel-AP*. Informationen zum Erstellen von Standardzugriffspunkten finden Sie unter [the section called "Erstellen von Zugriffspunkten"](#).
3. Führen Sie eine der folgenden Aktionen aus:
 - Erstellen Sie in Ihrem Konto eine Lambda-Funktion, mit der Sie Ihr Amazon-S3-Objekt transformieren möchten. Weitere Informationen zum Erstellen von Lambda-Funktionen finden Sie unter [the section called "Schreiben von Lambda-Funktionen"](#). Um Ihre benutzerdefinierte Funktion mit dem zu verwenden AWS CLI, siehe [Verwenden von Lambda mit AWS CLI](#) in der [AWS Lambda Leitfaden für Entwickler](#).
 - Verwenden Sie ein AWS vorgefertigte Lambda-Funktion. Weitere Hinweise zu vorgefertigten Funktionen finden Sie unter [Die Verwendung von AWS gebaute Lambda-Funktionen](#).
4. Erstellen Sie eine JSON Konfigurationsdatei mit dem Namen `my-object-lambda-configuration.json`. Geben Sie in dieser Konfiguration den unterstützenden Access Point und den Amazon-Ressourcennamen (ARN) für die Lambda-Funktion an, die Sie in den vorherigen Schritten erstellt haben, oder ARN für die vordefinierte Funktion, die Sie verwenden.

Example

```
{
  "SupportingAccessPoint" : "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
  "TransformationConfigurations": [{
    "Actions" : ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
    "ContentTransformation" : {
      "AwsLambda": {
        "FunctionPayload" : "{\"compressionType\":\"gzip\"}",
        "FunctionArn" : "arn:aws:lambda:us-east-1:111122223333:function/
compress"
      }
    }
  }]
}
```

5. Führen Sie den Befehl `create-access-point-for-object-lambda` aus, um den Object Lambda Access Point zu erstellen.

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --name my-object-lambda-ap --configuration file://my-olap-configuration.json
```

- (Optional) Erstellen Sie eine JSON Richtliniendatei mit dem Namen `my-olap-policy.json`

Durch Hinzufügen einer Ressourcenrichtlinie für Objekt-Lambda-Zugriffspunkte lässt sich die Verwendung des Zugriffspunkts nach Ressource, Benutzer oder anderen Bedingungen steuern. Diese Ressourcenrichtlinie gewährt die `GetObject` Berechtigung für das Konto `444455556666` zum angegebenen Object Lambda Access Point.

Example

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "Grant account 444455556666 GetObject access",
      "Effect": "Allow",
      "Action": "s3-object-lambda:GetObject",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
      },
      "Resource": "your-object-lambda-access-point-arn"
    }
  ]
}
```

- (Optional) Führen Sie den Befehl `put-access-point-policy-for-object-lambda` aus, um Ihre Ressourcenrichtlinie festzulegen.

```
aws s3control put-access-point-policy-for-object-lambda --account-id 111122223333 --name my-object-lambda-ap --policy file://my-olap-policy.json
```

- (Optional) Geben Sie eine Nutzlast an.

Eine Payload ist optionalJSON, die Sie Ihrem zur Verfügung stellen können AWS Lambda Funktion als Eingabe. Sie können Nutzlasten mit verschiedenen Parametern für verschiedene Object Lambda Access Points konfigurieren, die dieselbe Lambda-Funktion aufrufen, wodurch die Flexibilität Ihrer Lambda-Funktion erhöht wird.

Die folgende Konfiguration des Object Lambda Access Point zeigt eine Nutzlast mit zwei Parametern.

```
{
  "SupportingAccessPoint": "AccessPointArn",
  "CloudWatchMetricsEnabled": false,
  "TransformationConfigurations": [{
    "Actions": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
    "ContentTransformation": {
      "AwsLambda": {
        "FunctionArn": "FunctionArn",
        "FunctionPayload": "{\"res-x\": \"100\\\", \"res-y\": \"100\\\"}"
      }
    }
  ]
}
```

Die folgende Konfiguration des Object Lambda Access Point zeigt eine Nutzlast mit einem Parameter und den aktivierten Einstellungen GetObject-Range, GetObject-PartNumber, HeadObject-Range und HeadObject-PartNumber.

```
{
  "SupportingAccessPoint": "AccessPointArn",
  "CloudWatchMetricsEnabled": false,
  "AllowedFeatures": ["GetObject-Range", "GetObject-PartNumber", "HeadObject-Range", "HeadObject-PartNumber"],
  "TransformationConfigurations": [{
    "Action": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
    "ContentTransformation": {
      "AwsLambda": {
        "FunctionArn": "FunctionArn",
        "FunctionPayload": "{\"compression-amount\": \"5\\\"}"
      }
    }
  ]
}
```

⚠ Important

Wenn Sie Object Lambda Access Points verwenden, stellen Sie sicher, dass die Nutzlast keine vertraulichen Informationen enthält.

Verwendung der AWS CloudFormation Konsole und Vorlage

Sie können einen Object Lambda Access Point mithilfe der von Amazon S3 bereitgestellten Standardkonfiguration erstellen. Sie können eine heruntergeladene AWS CloudFormation Vorlage und Lambda-Funktions-Quellcode aus dem [GitHub Repository](#) und stellen Sie diese Ressourcen bereit, um einen funktionierenden Object Lambda Access Point einzurichten.

Informationen zum Ändern der AWS CloudFormation Die Standardkonfiguration der Vorlage finden Sie unter [the section called “Automatisieren Sie die Einrichtung von S3 Object Lambda mit AWS CloudFormation”](#).

Informationen zur Konfiguration von Object Lambda Access Points mithilfe von AWS CloudFormation ohne die Vorlage finden Sie [AWS::S3ObjectLambda::AccessPoint](#) in der AWS CloudFormation Benutzerleitfaden.


So laden Sie das Bereitstellungspaket für die Lambda-Funktion hoch

1. Laden Sie das heruntergeladene AWS Lambda Funktionsbereitstellungspaket `s3objectlambda_deployment_package.zip` in der [Standardkonfiguration von S3 Object Lambda](#).
2. Hochladen des Pakets in einen Amazon S3 Bucket

So erstellen Sie einen Object Lambda Access Point mit der AWS CloudFormation Konsole


1. Laden Sie das heruntergeladene AWS CloudFormation Vorlage `s3objectlambda_defaultconfig.yaml` in der [Standardkonfiguration von S3 Object Lambda](#).
2. Melden Sie sich an bei der AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
3. Führen Sie eine der folgenden Aktionen aus:

- Wenn Sie es noch nie benutzt haben AWS CloudFormation vorher, auf dem AWS CloudFormation Startseite, wählen Sie Stapel erstellen.
 - Wenn Sie verwendet haben AWS CloudFormation Wählen Sie zuvor im linken Navigationsbereich Stacks aus. Wählen Sie Create stack (Stack erstellen) und dann With new resources (standard) (Mit neuen Ressourcen (Standard)) aus.
4. Wählen Sie für Voraussetzung - Vorlage vorbereiten die Option Vorlage ist bereit aus.
 5. Wählen Sie unter Specify template (Vorlage angeben) die Option Upload a template file (Vorlagendatei hochladen) aus und laden Sie `s3objectlambda_defaultconfig.yaml` hoch.
 6. Wählen Sie Weiter.
 7. Geben Sie auf der Seite Stackdetails angeben einen Namen für den Stack ein.
 8. Geben Sie im Abschnitt Parameters (Parameter) die folgenden Parameter an, die in der Stack-Vorlage definiert sind:
 - a. Führen Sie für `CreateNewSupportingAccessPoint` einen der folgenden Schritte aus:
 - Wenn Sie bereits einen unterstützenden Zugriffspunkt für den S3-Bucket haben, in den Sie die Vorlage hochgeladen haben, wählen Sie `false` (falsch) aus.
 - Wenn Sie einen neuen Zugriffspunkt für diesen Bucket erstellen möchten, wählen Sie `true` (wahr) aus.
 - b. Wählen Sie für `EnableCloudWatchMonitoring` wahr oder falsch, je nachdem, ob Sie CloudWatch Amazon-Anforderungsmetriken und Alarme aktivieren möchten.
 - c. (Optional) Fügen Sie JSON Text hinzu `LambdaFunctionPayload`, den Sie Ihrer Lambda-Funktion als Eingabe zur Verfügung stellen möchten. Sie können Nutzlasten mit verschiedenen Parametern für verschiedene Object Lambda Access Points konfigurieren, die dieselbe Lambda-Funktion aufrufen, wodurch die Flexibilität Ihrer Lambda-Funktion erhöht wird.
 - d. Geben Sie für `LambdaFunctionRuntime` Ihre bevorzugte Laufzeit für die Lambda-Funktion ein. Die verfügbaren Auswahlmöglichkeiten sind `nodejs14.x`, `python3.9`, `java11`.

 **Important**

Wenn Sie Object Lambda Access Points verwenden, stellen Sie sicher, dass die Nutzlast keine vertraulichen Informationen enthält.

- e. Geben Sie für LambdaFunctionS3 BucketName den Amazon S3 S3-Bucket-Namen ein, in den Sie das Bereitstellungspaket hochgeladen haben.
- f. Geben Sie für LambdaFunctionS3Key den Amazon S3 S3-Objektschlüssel ein, in den Sie das Bereitstellungspaket hochgeladen haben.
- g. Geben Sie für LambdaFunctionS3 die Amazon S3 S3-Objektversion einObjectVersion, in die Sie das Bereitstellungspaket hochgeladen haben.
- h. Geben Sie für ObjectLambdaAccessPointName einen Namen für Ihren Object Lambda Access Point ein.
- i. Geben Sie für S3 BucketName den Amazon S3 S3-Bucket-Namen ein, der Ihrem Object Lambda Access Point zugeordnet wird.
- j. Geben Sie für SupportingAccessPointName den Namen Ihres unterstützenden Access Points ein.

 Note

Dies ist ein Zugriffspunkt, der mit dem Amazon-S3-Bucket verknüpft ist, den Sie im vorherigen Schritt ausgewählt haben. Wenn Ihrem Amazon S3 S3-Bucket keine Access Points zugeordnet sind, können Sie die Vorlage so konfigurieren, dass sie einen für Sie erstellt, indem Sie `true` für `CreateNewSupportingAccessPoint` wählen.

9. Wählen Sie Weiter.
10. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.

Weitere Informationen zu den optionalen Einstellungen auf dieser Seite finden Sie unter [Einstellung AWS CloudFormation Stapeloptionen](#) in der AWS CloudFormation Benutzerleitfaden.

11. Wählen Sie auf der Seite Prüfen Stack erstellen aus.

Verwendung der AWS Cloud Development Kit (AWS CDK)

Weitere Informationen zur Konfiguration von Object Lambda Access Points mit dem AWS CDK, siehe [AWS::S3ObjectLambdaConstruct Library](#) im AWS Cloud Development Kit (AWS CDK) APIReferenz.

Automatisieren Sie die Einrichtung von S3 Object Lambda mit einer Vorlage CloudFormation

Sie können eine verwenden AWS CloudFormation Vorlage zur schnellen Erstellung eines Amazon S3 Object Lambda Access Points. Die CloudFormation Vorlage erstellt automatisch relevante Ressourcen und konfiguriert AWS Identity and Access Management (IAM) Rollen und richtet eine ein AWS Lambda Funktion, die Anfragen automatisch über den Object Lambda Access Point bearbeitet. Mit der CloudFormation Vorlage können Sie bewährte Verfahren implementieren, Ihre Sicherheitslage verbessern und Fehler reduzieren, die durch manuelle Prozesse verursacht werden.

Dieses [GitHub Repository](#) enthält die CloudFormation Vorlage und den Quellcode der Lambda-Funktion. Anweisungen zur Verwendung der Vorlage finden Sie unter [the section called “Erstellen von Objekt-Lambda-Zugriffspunkten”](#).

Die in der Vorlage bereitgestellte Lambda-Funktion führt keine Transformation aus. Stattdessen gibt sie Ihre Objekte im Ist-Zustand aus Ihrem S3-Bucket zurück. Sie können die Funktion klonen und Ihren eigenen Transformationscode hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Weitere Informationen zum Anpassen Ihrer Funktion finden Sie unter [the section called “Ändern der Lambda-Funktion”](#) und [the section called “Schreiben von Lambda-Funktionen”](#).

Ändern der Vorlage

Erstellen eines neuen unterstützenden Zugriffspunkts

S3 Object Lambda verwendet zwei Zugriffspunkte, einen Object Lambda Access Point und einen standardmäßigen S3-Zugriffspunkt, der als unterstützender Zugriffspunkt bezeichnet wird. Wenn Sie eine Anforderung an einen Object Lambda Access Point ausführen, ruft S3 entweder Lambda in Ihrem Namen auf oder delegiert die Anforderung an den unterstützenden Zugriffspunkt, abhängig von der Konfiguration von S3 Object Lambda. Sie können einen neuen unterstützenden Zugriffspunkt erstellen, indem Sie den folgenden Parameter als Teil des `aws cloudformation deploy`-Befehls, der beim Bereitstellen der Vorlage erstellt werden soll, übergeben.

```
CreateNewSupportingAccessPoint=true
```

Konfigurieren einer Nutzlast-Funktion

Sie können eine Nutzlast konfigurieren, um der Lambda-Funktion zusätzliche Daten bereitzustellen, indem Sie beim Bereitstellen der Vorlage den folgenden Parameter als Teil des `aws cloudformation deploy`-Befehls übergeben.


```
LambdaFunctionPayload="format=json"
```

CloudWatch Amazon-Überwachung aktivieren

Sie können die CloudWatch Überwachung aktivieren, indem Sie bei der Bereitstellung der Vorlage den folgenden Parameter als Teil des `aws cloudformation deploy` Befehls übergeben.

```
EnableCloudWatchMonitoring=true
```

Dieser Parameter aktiviert die Anforderungsmetriken Ihres Object Lambda Access Point für Amazon S3 und erstellt zwei CloudWatch Alarmer zur Überwachung von clientseitigen und serverseitigen Fehlern.

Note

Für die CloudWatch Nutzung durch Amazon fallen zusätzliche Kosten an. Weitere Informationen zu Amazon-S3-Anforderungsmetriken finden Sie unter [Überwachen und Protokollieren von Zugriffspunkten](#).

Details zu den Preisen finden Sie unter [CloudWatch -Preise](#).

Konfigurieren von Provisioned Concurrency

Zur Reduzierung der Latenz können Sie bereitgestellte Parallelität für die Lambda-Funktion konfigurieren, die den Object Lambda Access Point unterstützt, indem Sie die Vorlage so bearbeiten, dass sie die folgenden Zeilen unter `Resources` einschließt.

```
LambdaFunctionVersion:  
  Type: AWS::Lambda::Version  
  Properties:  
    FunctionName: !Ref LambdaFunction  
    ProvisionedConcurrencyConfig:  
      ProvisionedConcurrentExecutions: Integer
```

Note

Für die Bereitstellung von Nebenläufigkeit fallen zusätzliche Gebühren an. Weitere Informationen zur bereitgestellten Parallelität finden Sie unter [Verwaltung der von Lambda bereitgestellten Parallelität](#) in der AWS Lambda Leitfaden für Entwickler.

Preisdetails finden Sie unter [AWS Lambda Preisgestaltung](#).

Ändern der Lambda-Funktion

Ändern der Header-Werte für eine **GetObject**-Anforderung

Standardmäßig leitet die Lambda-Funktion alle Header außer Content-Length und ETag von der vorsignierten URL Anfrage an den Client weiter. GetObject Basierend auf Ihrem Transformationscode in der Lambda-Funktion können Sie wählen, ob Sie neue Header-Werte an den GetObject-Client senden.

Sie können Ihre Lambda-Funktion aktualisieren, um neue Header-Werte zu senden, indem Sie sie in der WriteGetObjectResponse API Operation übergeben.

Wenn Ihre Lambda-Funktion beispielsweise den Text in Amazon-S3-Objekten in eine andere Sprache übersetzt, können Sie einen neuen Wert im Content-Language-Header bestätigen.

Ändern Sie die writeResponse-Funktion dazu wie folgt:

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
  transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    ContentLanguage: 'my-new-language'
  }).promise();
}
```

Eine vollständige Liste der unterstützten Header finden Sie [WriteGetObjectResponse](#) in der Amazon Simple Storage Service API Reference.

Zurückgeben von Metadaten Headern

Sie können Ihre Lambda-Funktion aktualisieren, um neue Header-Werte zu senden, indem Sie sie in der [WriteGetObjectResponse](#) API-Operationsanforderung übergeben.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
transformedObject: Buffer,
headers: Headers): Promise<PromiseResult<{}>, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest,
      'my-new-header': 'my-new-value'
    },
    ...headers
  }).promise();
}
```

Einen neuen Statuscode zurückgeben

Sie können einen benutzerdefinierten Statuscode an den GetObject Client zurückgeben, indem Sie ihn in der [WriteGetObjectResponse](#) API-Betriebsanforderung übergeben.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
transformedObject: Buffer,
headers: Headers): Promise<PromiseResult<{}>, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    StatusCode: Integer
  }).promise();
}
```

```
}
```

Eine vollständige Liste der unterstützten Statuscodes finden Sie [WriteGetObjectResponse](#) in der Amazon Simple Storage Service API Reference.

Anwenden der Parameter **Range** und **partNumber** auf das Quellobjekt

Standardmäßig kann der von der CloudFormation Vorlage erstellte Object Lambda Access Point die `partNumber` Parameter `Range` und verarbeiten. Die Lambda-Funktion wendet den Bereich oder die Teilenummer an, die auf das transformierte Objekt angefordert wird. Dazu muss die Funktion das gesamte Objekt herunterladen und die Transformation ausführen. In einigen Fällen werden Ihre transformierten Objektbereiche möglicherweise genau Ihren Quellobjektbereichen zugeordnet. Dies bedeutet, dass das Anfordern des Bytebereichs A-B für Ihr Quellobjekt und das Ausführen der Transformation zum gleichen Ergebnis führen kann, wie das Anfordern des gesamten Objekts, das Ausführen der Transformation und das Zurückgeben des Bytebereichs A-B für das transformierte Objekt.

In solchen Fällen können Sie die Implementierung der Lambda-Funktion ändern, um den Bereich oder die Teilenummer direkt auf das Quellobjekt anzuwenden. Dieser Ansatz verbessert die allgemeine Funktionslatenz und den erforderlichen Speicher. Weitere Informationen finden Sie unter [the section called "Arbeitet mit Range and partNumber Header"](#).

Deaktivieren der Verarbeitung von **Range** und **partNumber**

Standardmäßig kann der von der CloudFormation Vorlage erstellte Object Lambda Access Point die `partNumber` Parameter `Range` und verarbeiten. Wenn Sie dieses Verhalten nicht benötigen, können Sie es deaktivieren, indem Sie die folgenden Zeilen aus der Vorlage entfernen.

AllowedFeatures:

- GetObject-Range
- GetObject-PartNumber
- HeadObject-Range
- HeadObject-PartNumber

Transformieren von großen Objekten

Standardmäßig verarbeitet die Lambda-Funktion das gesamte Objekt im Speicher, bevor es mit dem Streamen der Antwort auf S3 Object Lambda beginnen kann. Sie können die Funktion ändern, um die Antwort zu streamen, während sie die Transformation durchführt. Dies hilft, die Transformationslatenz

und die Speichergröße der Lambda-Funktion zu reduzieren. Eine Beispielimplementierung finden Sie unter [Beispiel für komprimierte Inhalte streamen](#).

Verwenden von Amazon S3 Objekt-Lambda-Zugriffspunkten

Sie können Anforderungen über Amazon S3 Object Lambda Access Points ebenso vornehmen wie über andere Zugriffspunkte. Weitere Informationen darüber, wie Sie Anforderungen über einen Zugriffspunkt vornehmen können, finden Sie unter [Verwenden von Zugriffspunkten](#). Sie können Anfragen über Object Lambda Access Points stellen, indem Sie die Amazon S3 S3-Konsole verwenden. AWS Command Line Interface (AWS CLI), AWS SDKs oder Amazon S3 RESTAPI.

Important

Die Amazon-Ressourcennamen (ARNs) für Object Lambda Access Points verwenden den Dienstnamen. `s3-object-lambda` Somit ARNs beginnt Object Lambda Access Point mit `arn:aws::s3-object-lambda`, was mit anderen Access Points verwendet wird `arn:aws::s3`, und nicht mit dem, was mit anderen Access Points verwendet wird.

So finden Sie den Lambda Access Point ARN für Ihr Objekt

Um einen Object Lambda Access Point mit dem AWS CLI or AWS SDKs, Sie müssen den Amazon-Ressourcennamen (ARN) des Object Lambda Access Points kennen. Die folgenden Beispiele zeigen, wie Sie den ARN für einen Object Lambda Access Point mithilfe der Amazon S3 S3-Konsole oder AWS CLI.

Verwenden der S3-Konsole

So finden Sie den Lambda Access Point ARN für Ihren Object mithilfe der Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie das Optionsfeld neben dem Object Lambda Access Point, dessen ARN Sie kopieren möchten.
4. Wählen Sie die Option CopyARN Kopieren aus.

Verwendung der AWS CLI

Um den Lambda Access Point ARN für Ihren Object zu finden, verwenden Sie den AWS CLI

1. Um eine Liste der Object Lambda Access Points abzurufen, die mit Ihrem verknüpft sind AWS-Konto, führen Sie den folgenden Befehl aus. Bevor Sie den Befehl ausführen, ersetzen Sie die Konto-ID **111122223333** mit deinem AWS-Konto AUSWEIS.

```
aws s3control list-access-points-for-object-lambda --account-id 111122223333
```

2. Überprüfen Sie die Befehlsausgabe, um den Object Lambda Access Point zu findenARN, den Sie verwenden möchten. Die Ausgabe des vorherigen Befehls sollte dem folgenden Beispiel gleichen.

```
{
  "ObjectLambdaAccessPointList": [
    {
      "Name": "my-object-lambda-ap",
      "ObjectLambdaAccessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap"
    },
    ...
  ]
}
```

So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets

Wenn Sie einen Object Lambda Access Point erstellen, generiert Amazon S3 automatisch einen eindeutigen Alias für den Object Lambda Access Point. Sie können diesen Alias anstelle eines Amazon S3 S3-Bucket-Namens oder des Object Lambda Access Point Amazon Resource Name (ARN) in einer Anfrage für Access Point-Datenebenenoperationen verwenden. Eine Liste dieser Vorgänge finden Sie unter [Kompatibilität des Zugangspunkts mit AWS service](#).

Ein Aliasname eines Object Lambda Access Point wird innerhalb desselben Namespace wie ein Amazon-S3-Bucket erstellt. Dieser Aliasname wird automatisch generiert und kann nicht geändert werden. Für einen vorhandenen Object Lambda Access Point wird automatisch ein Alias zur Verwendung zugewiesen. Ein Aliasname eines Object Lambda Access Point erfüllt alle Anforderungen eines gültigen Amazon-S3-Bucketnamens und besteht aus den folgenden Teilen:

Object Lambda Access Point name prefix-metadata--o1-s3

Note

Das Suffix `--o1-s3` ist für Aliasnamen von Object Lambda Access Points reserviert und kann nicht für die Bucket-Namen oder die Namen von Object Lambda Access Points verwendet werden. Weitere Informationen zu Amazon-S3-Bucket-Benennungsregeln finden Sie unter [Regeln für die Benennung von Buckets](#).

Die folgenden Beispiele zeigen den ARN und den Object Lambda Access Point-Alias für einen Object Lambda Access Point mit dem Namen: *my-object-lambda-access-point*

- ARN – `arn:aws:s3-object-lambda:region:account-id:accesspoint/my-object-lambda-access-point`
- Alias des Object Lambda Access Point – `my-object-lambda-acc-1a4n8yjr3kda96f67zwrwiuse1a--o1-s3`

Wenn Sie einen Object Lambda Access Point verwenden, können Sie den Aliasnamen des Object Lambda Access Point verwenden, ohne dass umfangreiche Codeänderungen erforderlich sind.

Wenn Sie einen Object Lambda Access Point löschen, wird der Aliasname des Object Lambda Access Point inaktiv und die Bereitstellung wird aufgehoben.

So finden Sie den Alias für Ihren Object Lambda Access Point

Verwenden der S3-Konsole

So finden Sie den Alias für Ihren Object Lambda Access Point mithilfe der Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Kopieren Sie den Wert unter Alias des Objekt-Lambda-Zugriffspunkts für den Object Lambda Access Point, den Sie verwenden möchten.

Verwendung der AWS CLI

Wenn Sie einen Object Lambda Access Point erstellen, generiert Amazon S3 automatisch einen Aliasnamen für den Object Lambda Access Point wie im folgenden Beispielbefehl gezeigt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen. Informationen zum Erstellen eines Object Lambda Access Points mit dem AWS CLI, finden Sie unter [So erstellen Sie einen Object Lambda Access Point mit dem AWS CLI](#).

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-access-point --configuration file://my-olap-configuration.json
{
  "ObjectLambdaAccessPointArn": "arn:aws:s3:region:111122223333:accesspoint/my-
access-point",
  "Alias": {
    "Value": "my-object-lambda-acc-1a4n8yjr3kda96f67zwrwiiuse1a--ol-s3",
    "Status": "READY"
  }
}
```

Der generierte Aliasname des Object Lambda Access Point besteht aus zwei Feldern:

- Das Feld `Value` ist der Aliaswert des Object Lambda Access Point.
- Das Feld `Status` ist der Status des Alias des Object Lambda Access Point. Wenn der Status `PROVISIONING` lautet, stellt Amazon S3 den Alias des Object Lambda Access Point bereit und der Alias kann noch nicht verwendet werden. Wenn der Status `READY` lautet, wurde der Alias des Object Lambda Access Point erfolgreich bereitgestellt und kann verwendet werden.

Weitere Informationen zum `ObjectLambdaAccessPointAlias` Datentyp in der finden Sie REST API unter [CreateAccessPointForObjectLambda](#) und [ObjectLambdaAccessPointAlias](#) in der Amazon Simple Storage Service API Reference.

So verwenden Sie den Alias des Object Lambda Access Point

Sie können einen Alias eines Object Lambda Access Point anstelle eines Amazon-S3-Bucketnamens für die unter [Kompatibilität des Zugangspunkts mit AWS service](#) aufgeführten Operationen verwenden.

Folgendes AWS CLI Beispiel für den `get-bucket-location` Befehl verwendet den Access Point-Alias des Buckets, um Folgendes zurückzugeben AWS-Region in dem sich der Bucket befindet. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.


```
aws s3api get-bucket-location --bucket my-object-lambda-acc-w7i37nq6xuzgax3jw3oqtifiusw2a--o1-s3
```

```
{  
  "LocationConstraint": "us-west-2"  
}
```

Wenn der Alias des Object Lambda Access Point in einer Anforderung nicht gültig ist, wird der Fehlercode `InvalidAccessPointAliasError` zurückgegeben. Weitere Informationen zu `InvalidAccessPointAliasError` finden Sie in [der Liste der Fehlercodes](#) in der Amazon Simple Storage Service API Reference.

Die Einschränkungen eines Alias eines Object Lambda Access Point sind die gleichen wie bei einem Zugriffspunkt-Alias. Weitere Informationen zu den Einschränkungen eines Zugriffspunkt-Alias finden Sie unter [Einschränkungen](#).

Sicherheitsüberlegungen für S3 Object Lambda-Zugriffspunkte

Mit Amazon S3 Object Lambda können Sie benutzerdefinierte Transformationen an Daten durchführen, wenn diese Amazon S3 verlassen, indem Sie den Umfang und die Flexibilität von nutzen AWS Lambda als Rechenplattform. S3 und Lambda bleiben standardmäßig sicher, aber besondere Rücksicht vom Lambda-Funktionsautor ist erforderlich, um diese Sicherheit zu gewährleisten. S3 Object Lambda erfordert, dass der gesamte Zugriff über authentifizierte Prinzipale (kein anonymer Zugriff) und mehr erfolgt. HTTPS

Es wird Folgendes empfohlen, um Sicherheitsrisiken zu minimieren:

- Beschränken Sie die Lambda-Ausführungsrolle auf den geringsten Satz von Berechtigungen.
- Stellen Sie nach Möglichkeit sicher, dass Ihre Lambda-Funktion über das bereitgestellte Presigned auf Amazon S3 zugreift. URL

Richtlinien konfigurieren IAM

Unterstützung für S3-Zugangspunkte AWS Identity and Access Management (IAM)

Ressourcenrichtlinien, mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Weitere Informationen finden Sie unter [Konfiguration von IAM Richtlinien für Object Lambda Access Points](#).

Verhalten der Verschlüsselung

Weil Object Lambda Access Points sowohl Amazon S3 als auch AWS Lambda, es gibt Unterschiede im Verschlüsselungsverhalten. Weitere Informationen zum standardmäßigen S3-Verschlüsselungsverhalten finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

- Wenn Sie die serverseitige Verschlüsselung in S3 mit Object Lambda Access Points verwenden, wird das Objekt entschlüsselt, bevor es an Lambda gesendet wird. Nachdem das Objekt an Lambda gesendet wurde, wird es unverschlüsselt verarbeitet (im Fall einer GET- oder HEAD-Anforderung).
- Um zu verhindern, dass der Verschlüsselungsschlüssel protokolliert wird, lehnt GET S3 HEAD Anfragen nach Objekten ab, die mithilfe einer serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verschlüsselt wurden. Die Lambda-Funktion kann diese Objekte jedoch weiterhin abrufen, sofern sie Zugriff auf den vom Client bereitgestellten Schlüssel hat.
- Wenn Sie die clientseitige Verschlüsselung in S3 mit Object Lambda Access Points verwenden, stellen Sie sicher, dass Lambda Zugriff auf den Verschlüsselungsschlüssel hat, damit es das Objekt entschlüsseln und erneut verschlüsseln kann.

Sicherheit für Zugriffspunkte

S3 Object Lambda verwendet zwei Zugriffspunkte, einen Object Lambda Access Point und einen standardmäßigen S3-Zugriffspunkt, der als unterstützender Zugriffspunkt bezeichnet wird. Wenn Sie eine Anforderung an einen Object Lambda Access Point ausführen, ruft S3 entweder Lambda in Ihrem Namen auf oder delegiert die Anforderung an den unterstützenden Zugriffspunkt, abhängig von der Konfiguration von S3 Object Lambda. Wenn Lambda für eine Anfrage aufgerufen wird, generiert S3 in Ihrem Namen über URL den unterstützenden Access Point ein Ihrem Objekt vorsigniertes Objekt. Ihre Lambda-Funktion erhält dies URL als Eingabe, wenn die Funktion aufgerufen wird.

Sie können Ihre Lambda-Funktion so einstellen, dass sie dieses vorsignierte Objekt verwendet, URL um das ursprüngliche Objekt abzurufen, anstatt S3 direkt aufzurufen. Mit diesem Modell können Sie bessere Sicherheitsgrenzen auf Ihre Objekte anwenden. Sie können den direkten Objektzugriff über S3-Buckets oder S3-Zugriffspunkte auf eine begrenzte Anzahl von IAM Rollen oder Benutzern beschränken. Dieser Ansatz schützt Ihre Lambda-Funktionen auch davor, dem [Problem des verwirrten Stellvertreters](#) ausgesetzt zu sein, bei dem eine falsch konfigurierte Funktion mit

anderen Berechtigungen als Ihr Aufrufer den Zugriff auf Objekte zulassen oder verweigern könnte, wenn dies nicht der Fall ist.

Öffentlicher Zugriff auf Objekt-Lambda-Zugriffspunkt

S3 Object Lambda erlaubt keinen anonymen oder öffentlichen Zugriff, da Amazon S3 Ihre Identität autorisieren muss, um eine S3-Object-Lambda-Anforderung abzuschließen. Wenn Sie Anforderungen über einen Object Lambda Access Point aufrufen, benötigen Sie die Berechtigung `lambda:InvokeFunction` für die konfigurierte Lambda-Funktion. Ebenso müssen Sie über die erforderlichen `s3:*` Berechtigungen verfügen, wenn Sie andere API Operationen über einen Object Lambda Access Point aufrufen.

Ohne diese Berechtigungen schlagen Anfragen zum Aufrufen von Lambda oder zum Delegieren an S3 mit HTTP 403-Fehlern (verboten) fehl. Jeder Zugriff muss von authentifizierten Prinzipalen erfolgen. Wenn Sie einen öffentlichen Zugang benötigen, können Sie `Lambda@Edge` als mögliche Alternative verwenden. Weitere Informationen finden Sie unter [Customizing at the Edge with Lambda@Edge](#) im Amazon CloudFront Developer Guide.

IP-Adressen von Object Lambda Access Points

Die `describe-managed-prefix-lists` Subnetze unterstützen Gateway-Endpunkte für virtuelle private Clouds (VPC) und beziehen sich auf die Routing-Tabelle der Endpunkte. VPC Da Object Lambda Access Point kein Gateway unterstützt, fehlen VPC seine IP-Bereiche. Die fehlenden Bereiche gehören zu Amazon S3, werden aber von VPC Gateway-Endpunkten nicht unterstützt. Weitere Informationen zu `describe-managed-prefix-lists` finden Sie [DescribeManagedPrefixLists](#) in der EC2API Amazon-Referenz und [AWS IP-Adressbereiche](#) in Allgemeine AWS-Referenz.

Konfiguration von IAM Richtlinien für Object Lambda Access Points

Unterstützung Amazon S3 S3-Zugriffspunkte AWS Identity and Access Management (IAM) Ressourcenrichtlinien, mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Sie können den Zugriff über eine optionale Ressourcenrichtlinie auf Ihrem Object Lambda Access Point oder eine Ressourcenrichtlinie für unterstützende Zugriffspunkte steuern. step-by-step Beispiele finden Sie unter [Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#) und [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#).

Die vier folgenden Ressourcen müssen über Berechtigungen verfügen, um mit Object Lambda Access Points zu arbeiten:

- Die IAM Identität, z. B. Benutzer oder Rolle. Weitere Informationen zu IAM Identitäten und bewährten Methoden finden Sie im Benutzerhandbuch unter [IAM Identitäten \(Benutzer, Benutzergruppen und Rollen\)](#). IAM
- Der Bucket und der zugehörige Standardzugriffspunkt Wenn Sie mit Object Lambda Access Point arbeiten, wird dieser Standardzugriffspunkt als unterstützender Zugriffspunkt bezeichnet.
- Der Object Lambda Access Point.
- Das Tool AWS Lambda Funktion.

Important

Bevor Sie Ihre Richtlinie speichern, stellen Sie sicher, dass Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge von behoben haben AWS Identity and Access Management Access Analyzer. IAM Access Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM [Richtliniengrammatik](#) und der [bewährten Methoden](#) zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten umsetzbare Empfehlungen, die Sie beim Erstellen von Richtlinien unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen.

Weitere Informationen zur Überprüfung von Richtlinien mithilfe von IAM Access Analyzer finden Sie unter [IAM Access Analyzer-Richtlinienuvalidierung](#) im IAM Benutzerhandbuch. Eine Liste der Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM [IAM Access Analyzer-Referenz zur Richtlinienüberprüfung](#).

In den nachstehenden Richtlinienbeispielen wird davon ausgegangen, dass Sie die folgenden Ressourcen haben:

- Ein Amazon S3 S3-Bucket mit dem folgenden Amazon-Ressourcennamen (ARN):

```
arn:aws:s3:::amzn-s3-demo-bucket1
```

- Ein Amazon S3 S3-Standardzugriffspunkt in diesem Bucket mit den folgenden EigenschaftenARN:

```
arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point
```

- Ein Object Lambda Access Point mit den folgenden Eigenschaften: ARN

```
arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap
```

- Importieren in S3; AWS Lambda Funktion mit dem Folgenden: ARN

```
arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction
```

Note

Wenn Sie eine Lambda-Funktion von Ihrem Konto aus verwenden, müssen Sie die spezifische Funktionsversion in Ihre Richtlinienerklärung aufnehmen. Im folgenden Beispiel wird ARN die Version wie folgt angegeben **1**:

```
arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:1
```

Lambda unterstützt das Hinzufügen von IAM Richtlinien zur Version \$LATEST nicht. Weitere Informationen zu Lambda-Funktionsversionen finden Sie unter [Lambda-Funktionsversionen](#) im AWS Lambda Entwicklerhandbuch.

Example – Bucket-Richtlinie zum Delegieren der Zugriffskontrolle an Standardzugriffspunkte

Das folgende Beispiel für eine S3-Bucket-Richtlinie delegiert die Zugriffskontrolle für einen Bucket an die Standardzugriffspunkte des Buckets. Diese Richtlinie ermöglicht Vollzugriff auf alle Zugriffspunkte, die dem Konto des Bucket-Eigentümers gehören. Somit wird der gesamte Zugriff auf diesen Bucket durch die an seine Zugriffspunkte angehängten Richtlinien gesteuert. Benutzer können nur über einen Zugriffspunkt aus dem Bucket lesen, was bedeutet, dass Operationen nur über Zugriffspunkte aufgerufen werden können. Weitere Informationen finden Sie unter [Delegieren der Zugangskontrolle an Zugriffspunkte](#).

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Principal" : { "AWS": "account-ARN" },
      "Action" : "*",
      "Resource" : [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ],
      "Condition": {
```

```

        "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
ID" }
      }
    ]
  }
}

```

Example — IAM Richtlinie, die einem Benutzer die erforderlichen Berechtigungen zur Verwendung eines Object Lambda Access Points gewährt

Die folgende IAM Richtlinie gewährt einem Benutzer Berechtigungen für die Lambda-Funktion, den Standard-Access Point und den Object Lambda Access Point.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLambdaInvocation",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:1",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowStandardAccessPointAccess",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    }
  ]
}

```

```
    ]
  }
}
},
{
  "Sid": "AllowObjectLambdaAccess",
  "Action": [
    "s3-object-lambda:Get*",
    "s3-object-lambda:List*"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-  
object-lambda-ap"
}
]
```

Aktivieren von Berechtigungen für Lambda-Ausführungsrollen

Wenn GET Anfragen an einen Object Lambda Access Point gestellt werden, benötigt Ihre Lambda-Funktion die Erlaubnis, Daten an den S3 Object Lambda Access Point zu senden. Diese Berechtigung wird durch Aktivieren der `s3-object-lambda:WriteGetObjectResponse`-Berechtigung für die Ausführungsrolle Ihrer Lambda-Funktion bereitgestellt. Sie können eine neue Ausführungsrolle erstellen oder eine vorhandene Rolle aktualisieren.

Note

Ihre Funktion benötigt die `s3-object-lambda:WriteGetObjectResponse`-Berechtigung nur, wenn Sie eine GET-Anforderung stellen.

Um eine Ausführungsrolle in der Konsole zu erstellen IAM

1. Öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Wählen Sie Rolle erstellen.
4. Wählen Sie unter Häufige Anwendungsfälle die Option Lambda aus.
5. Wählen Sie Weiter.

- Suchen Sie auf der Seite „Berechtigungen hinzufügen“ nach AWS verwaltete Richtlinie [AmazonS3ObjectLambdaExecutionRolePolicy](#), und aktivieren Sie dann das Kontrollkästchen neben dem Richtliniennamen.

Diese Richtlinie sollte die Aktion `s3-object-lambda:WriteGetObjectResponse` enthalten.

- Wählen Sie Weiter.
- Geben Sie auf der Seite Name, review, and create (Name, prüfen und erstellen) für Role name (Rollenname) **s3-object-lambda-role** ein.
- (Optional) Fügen Sie eine Beschreibung und Tags für diese Rolle hinzu.
- Wählen Sie Rolle erstellen.
- Wenden Sie die neu erstellte **s3-object-lambda-role** als Ausführungsrolle Ihrer Lambda-Funktion. Dies kann während oder nach der Erstellung der Lambda-Funktion in der Lambda-Konsole erfolgen.

Weitere Informationen zu Ausführungsrollen finden Sie unter [Lambda-Ausführungsrolle](#) in der AWS Lambda Leitfaden für Entwickler.

Verwenden von Kontextschlüsseln mit Object Lambda Access Points

S3 Object Lambda wertet Kontextschlüssel wie `s3-object-lambda:TlsVersion` oder `s3-object-lambda:AuthType` im Zusammenhang mit der Verbindung oder dem Signieren der Anfrage aus. Alle anderen Kontextschlüssel, wie z. B. `s3:prefix`, werden von Amazon S3 ausgewertet.

Unterstützung für Object Lambda Access Point CORS

Wenn S3 Object Lambda eine Anforderung von einem Browser empfängt oder die Anforderung einen `Origin-Header` enthält, fügt S3 Object Lambda immer das Header-Feld `"AllowedOrigins": "*"` hinzu.

Weitere Informationen finden Sie unter [Verwendung von ursprungsübergreifendem Ressourcenaustausch \(\) CORS](#).

Schreiben von Lambda-Funktionen für S3 Object Lambda Access Points

In diesem Abschnitt wird beschrieben, wie man schreibt AWS Lambda Funktionen zur Verwendung mit Amazon S3 Object Lambda Access Points.

Informationen zu den vollständigen end-to-end Verfahren für einige S3 Object Lambda-Aufgaben finden Sie im Folgenden:

- [Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#)
- [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#)
- [Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen](#)

Themen

- [Arbeiten mit GetObject-Anforderungen in Lambda](#)
- [Arbeiten mit HeadObject-Anforderungen in Lambda](#)
- [Arbeiten mit ListObjects-Anforderungen in Lambda](#)
- [Arbeiten mit ListObjectsV2-Anforderungen in Lambda](#)
- [Format und Verwendung des Ereigniskontexts](#)
- [Arbeite mit Range and partNumber Header](#)

Arbeiten mit **GetObject**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für `GetObject` konfiguriert ist. S3 Object Lambda umfasst den Amazon S3 API S3-Vorgang `WriteGetObjectResponse`, der es der Lambda-Funktion ermöglicht, dem Anrufer benutzerdefinierte Daten und Antwortheader bereitzustellen. `GetObject`

`WriteGetObjectResponse` bietet Ihnen eine umfassende Kontrolle über den Statuscode, die Antwort-Header und den Antworttext, basierend auf Ihren Verarbeitungsanforderungen. Sie können `WriteGetObjectResponse` verwenden, um mit dem gesamten transformierten Objekt, Teilen des transformierten Objekts oder anderen Antworten basierend auf dem Kontext Ihrer Anwendung zu antworten. Der folgende Abschnitt zeigt einzigartige Beispiele für die Verwendung des Vorgangs.

`WriteGetObjectResponse` API

- Beispiel 1: Antworten Sie mit dem HTTP Statuscode 403 (Forbidden)
- Beispiel 2: Reagieren Sie mit einem transformierten Bild
- Beispiel 3: Komprimierte Inhalte streamen

Beispiel 1: Antworten Sie mit dem HTTP Statuscode 403 (Forbidden)

Sie können dies verwenden `WriteGetObjectResponse`, um mit dem HTTP Statuscode 403 (Forbidden) zu antworten, basierend auf dem Inhalt des Objekts.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.io.ByteArrayInputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example1 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
    Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();

        // Check to see if the request contains all of the necessary information.
        // If it does not, send a 4XX response and a custom error code and message.
        // Otherwise, retrieve the object from S3 and stream it
        // to the client unchanged.
        var tokenIsNotPresent = !
event.getUserRequest().getHeaders().containsKey("requiredToken");
        if (tokenIsNotPresent) {
            s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withStatusCode(403)
                .withContentLength(0L).withInputStream(new
ByteArrayInputStream(new byte[0]))
                .withErrorCode("MissingRequiredToken")
                .withErrorMessage("The required token was not present in the
request."));
            return;
        }
    }
}
```

```
// Prepare the presigned URL for use and make the request to S3.
HttpClient httpClient = HttpClient.newBuilder().build();
var presignedResponse = httpClient.send(
    HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
    HttpResponse.BodyHandlers.ofInputStream());

// Stream the original bytes back to the caller.
s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
    .withRequestRoute(event.outputRoute())
    .withRequestToken(event.outputToken())
    .withInputStream(presignedResponse.body()));
}
}
```

Python

```
import boto3
import requests

def handler(event, context):
    s3 = boto3.client('s3')

    """
    Retrieve the operation context object from the event. This object indicates
    where the WriteGetObjectResponse request
    should be delivered and contains a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    The 'userRequest' object has information related to the user who made this
    'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    user_request_headers = event["userRequest"]["headers"]

    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    # Check for the presence of a 'CustomHeader' header and deny or allow based on
    that header.
    is_token_present = "SuperSecretToken" in user_request_headers
```

```
if is_token_present:
    # If the user presented our custom 'SuperSecretToken' header, we send the
    requested object back to the user.
    response = requests.get(s3_url)
    s3.write_get_object_response(RequestRoute=route, RequestToken=token,
    Body=response.content)
else:
    # If the token is not present, we send an error back to the user.
    s3.write_get_object_response(RequestRoute=route, RequestToken=token,
    StatusCode=403,
    ErrorCode="NoSuperSecretTokenFound", ErrorMessage="The request was not
    secret enough.")

# Gracefully exit the Lambda function.
return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
    where the WriteGetObjectResponse request
    // should be delivered and contains a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    // The 'userRequest' object has information related to the user who made this
    'GetObject' request to S3 Object Lambda.
    const { userRequest, getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Check for the presence of a 'CustomHeader' header and deny or allow based on
    that header.
    const isTokenPresent = Object
        .keys(userRequest.headers)
        .includes("SuperSecretToken");

    if (!isTokenPresent) {
        // If the token is not present, we send an error back to the user. The
        'await' in front of the request
```

```
    // indicates that we want to wait for this request to finish sending before
    moving on.
    await s3.writeGetObjectResponse({
      RequestRoute: outputRoute,
      RequestToken: outputToken,
      StatusCode: 403,
      ErrorCode: "NoSuperSecretTokenFound",
      ErrorMessage: "The request was not secret enough.",
    }).promise();
  } else {
    // If the user presented our custom 'SuperSecretToken' header, we send the
    requested object back to the user.
    // Again, note the presence of 'await'.
    const presignedResponse = await axios.get(inputS3Url);
    await s3.writeGetObjectResponse({
      RequestRoute: outputRoute,
      RequestToken: outputToken,
      Body: presignedResponse.data,
    }).promise();
  }

  // Gracefully exit the Lambda function.
  return { statusCode: 200 };
}
```

Beispiel 2: Reagieren Sie mit einem transformierten Bild

Wenn Sie eine Bildtransformation durchführen, stellen Sie möglicherweise fest, dass Sie alle Bytes des Quellobjekts benötigen, bevor Sie mit der Verarbeitung beginnen können. In diesem Fall gibt Ihre `WriteGetObjectResponse`-Anforderung das gesamte Objekt in einem Aufruf an die anfragende Anwendung zurück.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;
```

```
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.awt.Image;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example2 {

    private static final int HEIGHT = 250;
    private static final int WIDTH = 250;

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Prepare the presigned URL for use and make the request to S3.
        var presignedResponse = httpClient.send(
            HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
            HttpResponse.BodyHandlers.ofInputStream());

        // The entire image is loaded into memory here so that we can resize it.
        // Once the resizing is completed, we write the bytes into the body
        // of the WriteGetObjectResponse request.
        var originalImage = ImageIO.read(presignedResponse.body());
        var resizingImage = originalImage.getScaledInstance(WIDTH, HEIGHT,
Image.SCALE_DEFAULT);
        var resizedImage = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        resizedImage.createGraphics().drawImage(resizingImage, 0, 0, WIDTH, HEIGHT,
null);

        var baos = new ByteArrayOutputStream();
        ImageIO.write(resizedImage, "png", baos);

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
            .withRequestRoute(event.outputRoute())
            .withRequestToken(event.outputToken()))
```

```
        .withInputStream(new ByteArrayInputStream(baos.toByteArray())));
    }
}
```

Python

```
import boto3
import requests
import io
from PIL import Image

def handler(event, context):
    """
    Retrieve the operation context object from the event. This object indicates
    where the WriteGetObjectResponse request
    should be delivered and has a presigned URL in 'inputS3Url' where we can
    download the requested object from.
    The 'userRequest' object has information related to the user who made this
    'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    """
    In this case, we're resizing .png images that are stored in S3 and are
    accessible through the presigned URL
    'inputS3Url'.
    """
    image_request = requests.get(s3_url)
    image = Image.open(io.BytesIO(image_request.content))
    image.thumbnail((256,256), Image.ANTIALIAS)

    transformed = io.BytesIO()
    image.save(transformed, "png")

    # Send the resized image back to the client.
    s3 = boto3.client('s3')
    s3.write_get_object_response(Body=transformed.getvalue(), RequestRoute=route,
    RequestToken=token)
```

```
# Gracefully exit the Lambda function.  
return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');  
const axios = require('axios').default;  
const sharp = require('sharp');  
  
exports.handler = async (event) => {  
  const s3 = new S3();  
  
  // Retrieve the operation context object from the event. This object indicates  
  // where the WriteGetObjectResponse request  
  // should be delivered and has a presigned URL in 'inputS3Url' where we can  
  // download the requested object from.  
  const { getObjectContext } = event;  
  const { outputRoute, outputToken, inputS3Url } = getObjectContext;  
  
  // In this case, we're resizing .png images that are stored in S3 and are  
  // accessible through the presigned URL  
  // 'inputS3Url'.  
  const { data } = await axios.get(inputS3Url, { responseType: 'arraybuffer' });  
  
  // Resize the image.  
  const resized = await sharp(data)  
    .resize({ width: 256, height: 256 })  
    .toBuffer();  
  
  // Send the resized image back to the client.  
  await s3.writeGetObjectResponse({  
    RequestRoute: outputRoute,  
    RequestToken: outputToken,  
    Body: resized,  
  }).promise();  
  
  // Gracefully exit the Lambda function.  
  return { statusCode: 200 };  
}
```

Beispiel 3: Komprimierte Inhalte streamen

Beim Komprimieren von Objekten werden komprimierte Daten inkrementell erzeugt. Folglich können Sie Ihre `WriteGetObjectResponse`-Anforderung verwenden, um die komprimierten Daten zurückzugeben, sobald sie bereit sind. Wie in diesem Beispiel gezeigt, ist es nicht notwendig, die Länge der abgeschlossenen Transformation zu kennen.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example3 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
    Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Request the original object from S3.
        var presignedResponse = httpClient.send(
            HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
            HttpResponse.BodyHandlers.ofInputStream());

        // Consume the incoming response body from the presigned request,
        // apply our transformation on that data, and emit the transformed bytes
        // into the body of the WriteGetObjectResponse request as soon as they're
        ready.
        // This example compresses the data from S3, but any processing pertinent
        // to your application can be performed here.
        var bodyStream = new GZIPCompressingInputStream(presignedResponse.body());

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest())
```

```
        .withRequestRoute(event.outputRoute())
        .withRequestToken(event.outputToken())
        .withInputStream(bodyStream));
    }
}
```

Python

```
import boto3
import requests
import zlib
from botocore.config import Config

"""
A helper class to work with content iterators. Takes an iterator and compresses the
bytes that come from it. It
implements 'read' and '__iter__' so that the SDK can stream the response.
"""
class Compress:
    def __init__(self, content_iter):
        self.content = content_iter
        self.compressed_obj = zlib.compressobj()

    def read(self, _size):
        for data in self.__iter__():
            return data

    def __iter__(self):
        while True:
            data = next(self.content)
            chunk = self.compressed_obj.compress(data)
            if not chunk:
                break

            yield chunk

        yield self.compressed_obj.flush()

def handler(event, context):
    """
```

Setting the 'payload_signing_enabled' property to False allows us to send a streamed response back to the client.

in this scenario, a streamed response means that the bytes are not buffered into memory as we're compressing them, but instead are sent straight to the user.

```
"""
my_config = Config(
    region_name='eu-west-1',
    signature_version='s3v4',
    s3={
        "payload_signing_enabled": False
    }
)
s3 = boto3.client('s3', config=my_config)

"""

Retrieve the operation context object from the event. This object indicates
where the WriteGetObjectResponse request
should be delivered and has a presigned URL in 'inputS3Url' where we can
download the requested object from.
The 'userRequest' object has information related to the user who made this
'GetObject' request to S3 Object Lambda.
"""

get_context = event["getObjectContext"]
route = get_context["outputRoute"]
token = get_context["outputToken"]
s3_url = get_context["inputS3Url"]

# Compress the 'get' request stream.
with requests.get(s3_url, stream=True) as r:
    compressed = Compress(r.iter_content())

# Send the stream back to the client.
s3.write_get_object_response(Body=compressed, RequestRoute=route,
RequestToken=token, ContentType="text/plain",
                             ContentEncoding="gzip")

# Gracefully exit the Lambda function.
return {'status_code': 200}
```

Node.js

```
const { S3 } = require('aws-sdk');
```

```
const axios = require('axios').default;
const zlib = require('zlib');

exports.handler = async (event) => {
  const s3 = new S3();

  // Retrieve the operation context object from the event. This object indicates
  // where the WriteGetObjectResponse request
  // should be delivered and has a presigned URL in 'inputS3Url' where we can
  // download the requested object from.
  const { getObjectContext } = event;
  const { outputRoute, outputToken, inputS3Url } = getObjectContext;

  // Download the object from S3 and process it as a stream, because it might be a
  // huge object and we don't want to
  // buffer it in memory. Note the use of 'await' because we want to wait for
  // 'writeGetObjectResponse' to finish
  // before we can exit the Lambda function.
  await axios({
    method: 'GET',
    url: inputS3Url,
    responseType: 'stream',
  }).then(
    // Gzip the stream.
    response => response.data.pipe(zlib.createGzip())
  ).then(
    // Finally send the gzip-ed stream back to the client.
    stream => s3.writeGetObjectResponse({
      RequestRoute: outputRoute,
      RequestToken: outputToken,
      Body: stream,
      ContentType: "text/plain",
      ContentEncoding: "gzip",
    }).promise()
  );

  // Gracefully exit the Lambda function.
  return { statusCode: 200 };
}
```

 Note

Obwohl S3 Object Lambda ermöglicht, bis zu 60 Sekunden eine vollständige Antwort an den Aufrufer über die `WriteGetObjectResponse`-Anforderung zu senden, ist die tatsächlich verfügbare Zeit möglicherweise geringer. Zum Beispiel kann Ihr Timeout für Lambda-Funktionen weniger als 60 Sekunden betragen. In anderen Fällen kann der Aufrufer strengere Zeitbeschränkungen haben.

Damit der ursprüngliche Anrufer eine andere Antwort als HTTP den Statuscode 500 (Interner Serverfehler) erhält, muss der `WriteGetObjectResponse` Anruf abgeschlossen sein. Wenn die Lambda-Funktion mit einer Ausnahme oder auf andere Weise zurückkehrt, bevor die `WriteGetObjectResponse` API Operation aufgerufen wird, erhält der ursprüngliche Aufrufer eine Antwort von 500 (Internal Server Error). Ausnahmen, die während der Zeit ausgelöst werden, die zum Abschließen der Antwort benötigt wird, führen zu verkürzten Antworten an den Aufrufer. Wenn die Lambda-Funktion vom `WriteGetObjectResponse` API Anruf eine Antwort mit dem HTTP Statuscode 200 (OK) erhält, hat der ursprüngliche Aufrufer die vollständige Anfrage gesendet. Die Antwort der Lambda-Funktion, wird von S3 Object Lambda, unabhängig davon, ob eine Ausnahme ausgelöst wird oder nicht, ignoriert.

Beim Aufrufen des `WriteGetObjectResponse` API Vorgangs benötigt Amazon S3 das Route- und Anforderungstoken aus dem Ereigniskontext. Weitere Informationen finden Sie unter [Format und Verwendung des Ereigniskontexts](#).

Die Routen- und Anforderungstokenparameter sind erforderlich, um die `WriteGetObjectResult`-Antwort mit dem ursprünglichen Aufrufer zu verbinden. Obwohl es immer angemessen ist, 500 Antworten (Interner Serverfehler) erneut zu versuchen, da es sich bei dem Anforderungstoken um ein einmalig verwendbares Token handelt, können nachfolgende Versuche, es zu Antworten mit dem HTTP Statuscode 400 (Bad Request) führen. Obwohl der Aufruf an `WriteGetObjectResponse` mit den Routen- und Anforderungstoken nicht von der aufgerufenen Lambda-Funktion erfolgen muss, ist hierfür eine Identität im selben Konto erforderlich. Der Aufruf muss außerdem abgeschlossen sein, bevor die Lambda-Funktion die Ausführung beendet.

Arbeiten mit **HeadObject**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für `HeadObject` konfiguriert ist. Lambda erhält eine JSON Nutzlast, die einen Schlüssel namens `headObjectContext` Innerhalb des Kontextes wird eine einzelne

Eigenschaft `inputS3Url`, die URL für den unterstützenden Zugriffspunkt vorsigniert ist.
`HeadObject`

Die vorsignierte Eigenschaft enthält URL die folgenden Eigenschaften, sofern sie angegeben sind:

- `versionId` (in den Abfrageparametern)
- `requestPayer` (im `x-amz-request-payer-Header`)
- `expectedBucketOwner` (im `x-amz-expected-bucket-owner-Header`)

Andere Eigenschaften werden nicht vorsigniert und daher nicht berücksichtigt. Nicht signierte Optionen, die als Header gesendet werden, können manuell zur Anfrage hinzugefügt werden, wenn die vorsignierte Option aufgerufen wird URL, die sich in den Headern befindet. `userRequest` Serverseitige Verschlüsselungsoptionen werden für `HeadObject` nicht unterstützt.

Informationen zu den URI Anforderungssyntaxparametern finden Sie [HeadObject](#) in der Amazon Simple Storage Service API Reference.

Das folgende Beispiel zeigt eine JSON Lambda-Eingabenutzlast für `HeadObject`

```
{
  "xAmzRequestId": "requestId",
  "**headObjectContext**": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>"
  },
  "configuration": {
    "accessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/example-object-lambda-ap",
    "supportingAccessPointArn": "arn:aws:s3:us-east-1:111122223333:accesspoint/example-ap",
    "payload": "{}"
  },
  "userRequest": {
    "url": "https://object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com/example",
    "headers": {
      "Host": "object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com",
      "Accept-Encoding": "identity",
      "X-Amz-Content-SHA256": "e3b0c44298fc1example"
    }
  }
}
```

```
},
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "principalId",
  "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
  "accountId": "111122223333",
  "accessKeyId": "accessKeyId",
  "sessionContext": {
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
    }
  },
  "sessionIssuer": {
    "type": "Role",
    "principalId": "principalId",
    "arn": "arn:aws:iam::111122223333:role/Admin",
    "accountId": "111122223333",
    "userName": "Admin"
  }
}
},
"protocolVersion": "1.00"
}
```

Ihre Lambda-Funktion sollte ein JSON Objekt zurückgeben, das die Header und Werte enthält, die für den `HeadObject` Aufruf zurückgegeben werden.

Das folgende Beispiel zeigt die Struktur der Lambda-Antwort JSON für `HeadObject`.

```
{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
  "headers": {
    "Accept-Ranges": <string>,
    "x-amz-archive-status": <string>,
    "x-amz-server-side-encryption-bucket-key-enabled": <boolean>,
    "Cache-Control": <string>,
    "Content-Disposition": <string>,
    "Content-Encoding": <string>,
    "Content-Language": <string>,
    "Content-Length": <number>, // Required
    "Content-Type": <string>,
```

```

    "x-amz-delete-marker": <boolean>,
    "ETag": <string>,
    "Expires": <string>,
    "x-amz-expiration": <string>,
    "Last-Modified": <string>,
    "x-amz-missing-meta": <number>,
    "x-amz-object-lock-mode": <string>,
    "x-amz-object-lock-legal-hold": <string>,
    "x-amz-object-lock-retain-until-date": <string>,
    "x-amz-mp-parts-count": <number>,
    "x-amz-replication-status": <string>,
    "x-amz-request-charged": <string>,
    "x-amz-restore": <string>,
    "x-amz-server-side-encryption": <string>,
    "x-amz-server-side-encryption-customer-algorithm": <string>,
    "x-amz-server-side-encryption-aws-kms-key-id": <string>,
    "x-amz-server-side-encryption-customer-key-MD5": <string>,
    "x-amz-storage-class": <string>,
    "x-amz-tagging-count": <number>,
    "x-amz-version-id": <string>,
    <x-amz-meta-headers>: <string>, // user-defined metadata
    "x-amz-meta-meta1": <string>, // example of the user-defined metadata header,
it will need the x-amz-meta prefix
    "x-amz-meta-meta2": <string>
    ...
};
}

```

Das folgende Beispiel zeigt, wie Sie Presigned verwenden, um Ihre Antwort aufzufüllen URL, indem Sie die Header-Werte nach Bedarf ändern, bevor Sie die zurückgeben. JSON

Python

```

import requests

def lambda_handler(event, context):
    print(event)

    # Extract the presigned URL from the input.
    s3_url = event["headObjectContext"]["inputS3Url"]

    # Get the head of the object from S3.
    response = requests.head(s3_url)

```



```
# Return the error to S3 Object Lambda (if applicable).
if (response.status_code >= 400):
    return {
        "statusCode": response.status_code,
        "errorCode": "RequestFailure",
        "errorMessage": "Request to S3 failed"
    }

# Store the headers in a dictionary.
response_headers = dict(response.headers)

# This obscures Content-Type in a transformation, it is optional to add
response_headers["Content-Type"] = ""

# Return the headers to S3 Object Lambda.
return {
    "statusCode": response.status_code,
    "headers": response_headers
}
```

Arbeiten mit **ListObjects**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für ListObjects konfiguriert ist. Lambda empfängt die JSON Nutzlast mit einem neuen Objekt namens `listObjectsContext`. `listObjectsContext` enthält eine einzelne Eigenschaft `inputS3Url`, die URL für den unterstützenden Access Point vorsigniert ist.

ListObjects

Im Gegensatz zu `GetObject` und `HeadObject` enthält das vorsignierte URL Objekt die folgenden Eigenschaften, wenn sie angegeben sind:

- Alle Abfrageparameter
- `requestPayer` (im `x-amz-request-payer-Header`)
- `expectedBucketOwner` (im `x-amz-expected-bucket-owner-Header`)

Informationen zu den URI Anforderungssyntaxparametern finden Sie [ListObjects](#) in der Amazon Simple Storage Service API Reference.

⚠ Important

Wir empfehlen, bei der Entwicklung von Anwendungen die neuere Version [ListObjectsV2](#) zu verwenden. Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin ListObjects.

Das folgende Beispiel zeigt die JSON Lambda-Eingabenutzlast für ListObjects

```
{
  "xAmzRequestId": "requestId",
  "***listObjectsContext***": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/?X-Amz-Security-Token=<snip>",
  },
  "configuration": {
    "accessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/example-object-lambda-ap",
    "supportingAccessPointArn": "arn:aws:s3:us-east-1:111122223333:accesspoint/example-ap",
    "payload": "{}"
  },
  "userRequest": {
    "url": "https://object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com/example",
    "headers": {
      "Host": "object-lambda-111122223333.s3-object-lambda.us-east-1.amazonaws.com",
      "Accept-Encoding": "identity",
      "X-Amz-Content-SHA256": "e3b0c44298fc1example"
    }
  },
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
    "accountId": "111122223333",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
      }
    }
  }
}
```

```
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "principalId",
      "arn": "arn:aws:iam::111122223333:role/Admin",
      "accountId": "111122223333",
      "userName": "Admin"
    }
  },
  "protocolVersion": "1.00"
}
```

Ihre Lambda-Funktion sollte ein JSON Objekt zurückgeben, das den Statuscode, das XML Listenergebnis oder die Fehlerinformationen enthält, die von S3 Object Lambda zurückgegeben werden.

S3 Object Lambda verarbeitet oder validiert `listResultXml` nicht, sondern leitet es an den `ListObjects`-Aufrufer weiter. Für `listBucketResult` erwartet S3 Object Lambda, dass bestimmte Eigenschaften von einem speziellen Typ sind, und löst Ausnahmen aus, wenn es sie nicht analysieren kann. `listResultXml` und `listBucketResult` können nicht gleichzeitig bereitgestellt werden.

Das folgende Beispiel zeigt, wie Sie das Presigned verwendenURL, um Amazon S3 aufzurufen und das Ergebnis zu verwenden, um eine Antwort zu füllen, einschließlich Fehlerprüfung.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsContext"]["inputS3Url"]

    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
```

```
    return {
        "statusCode": response.status_code,
        "errorCode": error["Error"]["Code"],
        "errorMessage": error["Error"]["Message"]
    }

# Store the XML result in a dict.
response_dict = xmltodict.parse(response.content)

# This obscures StorageClass in a transformation, it is optional to add
for item in response_dict['ListBucketResult']['Contents']:
    item['StorageClass'] = ""

# Convert back to XML.
listResultXml = xmltodict.unparse(response_dict)

# Create response with listResultXml.
response_with_list_result_xml = {
    'statusCode': 200,
    'listResultXml': listResultXml
}

# Create response with listBucketResult.
response_dict['ListBucketResult'] =
sanitize_response_dict(response_dict['ListBucketResult'])
response_with_list_bucket_result = {
    'statusCode': 200,
    'listBucketResult': response_dict['ListBucketResult']
}

# Return the list to S3 Object Lambda.
# Can return response_with_list_result_xml or response_with_list_bucket_result
return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
```

```

        newlist.append(element)
    value = newlist
elif type(value) == dict:
    value = sanitize_response_dict(value)
    new_response_dict[new_key] = value
return new_response_dict

```

Das folgende Beispiel zeigt die Struktur der Lambda-Antwort JSON für `ListObjects`.

```

{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
  "listResultXml": <string>; // This can also be Error XML string in case S3 returned
error response when calling the pre-signed URL

  "listBucketResult": { // listBucketResult can be provided instead of listResultXml,
however they can not both be provided in the JSON response
    "name": <string>, // Required for 'listBucketResult'
    "prefix": <string>,
    "marker": <string>,
    "nextMarker": <string>,
    "maxKeys": <int>, // Required for 'listBucketResult'
    "delimiter": <string>,
    "encodingType": <string>
    "isTruncated": <boolean>, // Required for 'listBucketResult'
    "contents": [ {
      "key": <string>, // Required for 'content'
      "lastModified": <string>,
      "eTag": <string>,
      "checksumAlgorithm": <string>, // CRC32, CRC32C, SHA1, SHA256
      "size": <int>, // Required for 'content'
      "owner": {
        "displayName": <string>, // Required for 'owner'
        "id": <string>, // Required for 'owner'
      },
      "storageClass": <string>
    },
    ...
  ],
  "commonPrefixes": [ {
    "prefix": <string> // Required for 'commonPrefix'

```

```

    },
    ...
  ],
}
}

```

Arbeiten mit **ListObjectsV2**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für ListObjectsV2 konfiguriert ist. Lambda empfängt die JSON Nutzlast mit einem neuen Objekt namens `listObjectsV2Context`. `listObjectsV2Context` enthält eine einzelne Eigenschaft `inputS3Url`, die URL für den unterstützenden Access Point vorsigniert ist.

Im Gegensatz zu `GetObject` und `HeadObject` enthält das Presigned URL die folgenden Eigenschaften, sofern sie angegeben sind:

- Alle Abfrageparameter
- `requestPayer` (im `x-amz-request-payer`-Header)
- `expectedBucketOwner` (im `x-amz-expected-bucket-owner`-Header)

Informationen zu den URI Anforderungssyntaxparametern finden Sie [ListObjectsV2](#) in der Amazon Simple Storage Service API Reference.

Das folgende Beispiel zeigt die JSON Lambda-Eingabenutzlast für `ListObjectsV2`

```

{
  "xAmzRequestId": "requestId",
  "**listObjectsV2Context**": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/?list-type=2&X-Amz-Security-Token=<snip>",
  },
  "configuration": {
    "accessPointArn": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/example-object-lambda-ap",
    "supportingAccessPointArn": "arn:aws:s3:us-east-1:111122223333:accesspoint/example-ap",
    "payload": "{}"
  },
  "userRequest": {

```

```

    "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
    "headers": {
      "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
      "Accept-Encoding": "identity",
      "X-Amz-Content-SHA256": "e3b0c44298fc1example"
    }
  },
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
    "accountId": "111122223333",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "principalId",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      }
    }
  },
  "protocolVersion": "1.00"
}

```

Ihre Lambda-Funktion sollte ein JSON Objekt zurückgeben, das den Statuscode, das XML Listenergebnis oder die Fehlerinformationen enthält, die von S3 Object Lambda zurückgegeben werden.

S3 Object Lambda verarbeitet oder validiert `listResultXml` nicht, sondern leitet es an den `ListObjectsV2`-Aufrufer weiter. Für `listBucketResult` erwartet S3 Object Lambda, dass bestimmte Eigenschaften von einem speziellen Typ sind, und löst Ausnahmen aus, wenn es sie nicht analysieren kann. `listResultXml` und `listBucketResult` können nicht gleichzeitig bereitgestellt werden.

Das folgende Beispiel zeigt, wie Sie das Presigned verwendenURL, um Amazon S3 aufzurufen und das Ergebnis zu verwenden, um eine Antwort zu füllen, einschließlich Fehlerprüfung.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsV2Context"]["inputS3Url"]

    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)

    # Create response with listResultXml.
    response_with_list_result_xml = {
        'statusCode': 200,
        'listResultXml': listResultXml
    }

    # Create response with listBucketResult.
```



```

    response_dict['ListBucketResult'] =
    sanitize_response_dict(response_dict['ListBucketResult'])
    response_with_list_bucket_result = {
        'statusCode': 200,
        'listBucketResult': response_dict['ListBucketResult']
    }

    # Return the list to S3 Object Lambda.
    # Can return response_with_list_result_xml or response_with_list_bucket_result
    return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict

```

Das folgende Beispiel zeigt die Struktur der Lambda-Antwort JSON für `ListObjectsV2`.

```

{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "listResultXml": <string>; // This can also be Error XML string in case S3 returned
    error response when calling the pre-signed URL

    "listBucketResult": { // listBucketResult can be provided instead of
    listResultXml, however they can not both be provided in the JSON response
        "name": <string>, // Required for 'listBucketResult'
        "prefix": <string>,
        "startAfter": <string>,

```

```

    "continuationToken": <string>,
    "nextContinuationToken": <string>,
    "keyCount": <int>, // Required for 'listBucketResult'
    "maxKeys": <int>, // Required for 'listBucketResult'
    "delimiter": <string>,
    "encodingType": <string>
    "isTruncated": <boolean>, // Required for 'listBucketResult'
    "contents": [ {
        "key": <string>, // Required for 'content'
        "lastModified": <string>,
        "eTag": <string>,
        "checksumAlgorithm": <string>, // CRC32, CRC32C, SHA1, SHA256
        "size": <int>, // Required for 'content'
        "owner": {
            "displayName": <string>, // Required for 'owner'
            "id": <string>, // Required for 'owner'
        },
        "storageClass": <string>
    },
    ...
],
    "commonPrefixes": [ {
        "prefix": <string> // Required for 'commonPrefix'
    },
    ...
],
}
}

```

Format und Verwendung des Ereigniskontexts

Amazon S3 Object Lambda bietet Kontext zu der Anfrage, die gestellt wird, wenn diese an Ihren AWS Lambda Funktion. Nachstehend finden Sie eine Beispielanforderung: Beschreibungen der Felder folgen nach dem Beispiel.

```

{
    "xAmzRequestId": "requestId",
    "getObjectContext": {
        "inputS3Url": "https://my-s3-ap-111122223333.s3-accesspoint.us-east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>",
        "outputRoute": "io-use1-001",
        "outputToken": "OutputToken"
    },
}

```

```

"configuration": {
  "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
  "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
  "payload": "{}"
},
"userRequest": {
  "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
  "headers": {
    "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
    "Accept-Encoding": "identity",
    "X-Amz-Content-SHA256": "e3b0c44298fc1example"
  }
},
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "principalId",
  "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
  "accountId": "111122223333",
  "accessKeyId": "accessKeyId",
  "sessionContext": {
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "principalId",
      "arn": "arn:aws:iam::111122223333:role/Admin",
      "accountId": "111122223333",
      "userName": "Admin"
    }
  }
},
"protocolVersion": "1.00"
}

```

Die folgenden Felder sind in der Anforderung enthalten:

- `xAmzRequestId` – Die Amazon-S3-Anforderungs-ID für diese Anforderung. Wir empfehlen, diesen Wert zu protokollieren, um beim Debuggen zu helfen.
- `getObjectContext` – Die Eingabe- und Ausgabedetails für Verbindungen zu Amazon S3 und S3 Object Lambda.
 - `inputS3Url`— Eine vorsignierte URL, die verwendet werden kann, um das Originalobjekt von Amazon S3 abzurufen. Der URL ist mit der Identität des ursprünglichen Anrufers signiert, und die Berechtigungen dieses Benutzers gelten, wenn der URL verwendet wird. Wenn der signierte Header enthält URL, muss die Lambda-Funktion diese Header in den Aufruf von Amazon S3 einbeziehen, mit Ausnahme des Headers. `Host`
 - `outputRoute`— Ein Routing-Token, das dem S3 Object Lambda hinzugefügt wird, URL wenn die Lambda-Funktion aufgerufen wird. `WriteGetObjectResponse`
 - `outputToken` – Ein undurchsichtiges Token, das von S3 Object Lambda verwendet wird, um den `WriteGetObjectResponse`-Aufruf mit dem ursprünglichen Aufrufer abzugleichen.
- `configuration` – Informationen zur Konfiguration des Object Lambda Access Point.
 - `accessPointArn`— Der Amazon-Ressourcenname (ARN) des Object Lambda Access Points, der diese Anfrage erhalten hat.
 - `supportingAccessPointArn`— Der ARN des unterstützenden Access Points, der in der Object Lambda Access Point-Konfiguration angegeben ist.
 - `payload` – Benutzerdefinierte Daten, die auf die Konfiguration des Object Lambda Access Point angewendet werden. S3 Object Lambda behandelt diese Daten als eine undurchsichtige Zeichenfolge, daher muss sie möglicherweise vor der Verwendung dekodiert werden.
- `userRequest` – Informationen über den ursprünglichen Aufruf von S3 Object Lambda.
 - `url`— Die Dekodierung URL der Anfrage, wie sie von S3 Object Lambda empfangen wurde, mit Ausnahme aller autorisierungsbezogenen Abfrageparameter.
 - `headers`— Eine Zuordnung von Zeichenketten zu Zeichenketten, die die HTTP Header und ihre Werte aus dem ursprünglichen Aufruf enthalten, ausgenommen alle autorisierungsbezogenen Header. Wenn derselbe Header mehrfach erscheint, werden die Werte von jeder Instance desselben Header zu einer durch Kommata getrennten Liste zusammengefasst. Der Fall der ursprünglichen Header wird in dieser Zuordnung beibehalten.
- `userIdentity` – Details zur die Identität, die den Aufruf von S3 Object Lambda getätigt hat. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen](#) für Pfade in der AWS CloudTrail Benutzerleitfaden.
 - `type` – Die Art der Identität.
 - `accountId`— Das AWS-Konto zu dem die Identität gehört.

- `userName` – Der Anzeigename der Identität, von der der Aufruf stammt.
- `principalId` – Die eindeutige ID für die Entität, von der der Aufruf stammt.
- `arn`— Der ARN des Schulleiters, der den Anruf getätigt hat. Der letzte Abschnitt von ARN enthält den Benutzer oder die Rolle, die den Anruf getätigt hat.
- `sessionContext` – Erfolgte die Abfrage mittels temporärer Sicherheitsanmeldeinformationen, stellt dieses Element Informationen über die Sitzung bereit, das für diese Anmeldeinformationen erstellt wurde.
- `invokedBy`— Der Name des AWS-Service der die Anfrage gestellt hat, wie Amazon EC2 Auto Scaling oder AWS Elastic Beanstalk.
- `sessionIssuer` – Erfolgte die Abfrage mittels temporärer Sicherheitsanmeldeinformationen, gibt dieses Element Auskunft darüber, wie die Anmeldeinformationen erhalten wurden.
- `protocolVersion` – Die Versions-ID des bereitgestellten Kontextes. Das Format des Felds ist `{Major Version}.{Minor Version}`. Die Nebenversionsnummern sind immer zweistellige Zahlen. Jede Entfernung oder Änderung der Semantik eines Feldes erfordert einen Sprung der Hauptversion und erfordert ein aktives Opt-In. Amazon S3 kann jederzeit neue Felder hinzufügen. Zu diesem Zeitpunkt kann es zu einem Sprung der Nebenversion kommen. Aufgrund der Art der Software-Einführung sehen Sie möglicherweise die Verwendung mehrerer Nebenversionen gleichzeitig.

Arbeitet mit Range and partNumber Header

Wenn Sie in Amazon S3 Object Lambda mit großen Objekten arbeiten, können Sie den Range HTTP Header verwenden, um einen bestimmten Bytebereich von einem Objekt herunterzuladen. Sie können gleichzeitige Verbindungen zu Amazon S3 verwenden, um verschiedene Bytebereiche aus demselben Objekt abzurufen. Sie können auch den `partNumber`-Parameter (eine Ganzzahl zwischen 1 und 10 000) verwenden, der eine „ranged“ Anforderung für das angegebene Teil des Objekts ausführt.

Weil es mehrere Möglichkeiten gibt, eine Anforderung zu bearbeiten, die den Range- oder `partNumber`-Parameter beinhaltet, wendet S3 Object Lambda diese Parameter nicht auf das transformierte Objekt an. Stattdessen Ihr AWS Lambda Die Funktion muss diese Funktionalität nach Bedarf für Ihre Anwendung implementieren.

Wenn Sie die Parameter Range und `partNumber` mit S3 Object Lambda verwenden möchten, gehen Sie wie folgt vor:

- Aktivieren Sie diese Parameter in der Konfiguration Ihres Object Lambda Access Point.
- Schreiben Sie eine Lambda-Funktion, die Anforderungen verarbeiten kann, die diese Parameter enthalten.

In den folgenden Schritten wird beschrieben, wie Sie dies erreichen.

Schritt 1: Konfigurieren Ihres Object Lambda Access Point

Standardmäßig antworten Object Lambda Access Points mit dem HTTP Statuscode 501 (Nicht implementiert) auf jede `HeadObject` OR-Anfrage, die einen `GetObject partNumber` OR-Parameter enthält, entweder in den Headern `Range` oder in den Abfrageparametern.

Damit ein Object-Lambda-Zugriffspunkt solche Anforderungen annehmen kann, müssen Sie `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` oder `HeadObject-PartNumber` in den `AllowedFeatures`-Abschnitt die Konfiguration Ihres Object-Lambda-Zugangspunkts einschließen. Weitere Informationen zur Aktualisierung der Konfiguration Ihres Object Lambda Access Point finden Sie unter [Erstellen von Objekt-Lambda-Zugriffspunkten](#).

Schritt 2: Implementieren Sie das **Range** oder **partNumber** in Ihrer Lambda-Funktion

Wenn Ihr Object Lambda Access Point Ihre Lambda-Funktion mit einer `GetObject`- oder `HeadObject`-Bereichsanforderung aufruft, ist der `Range`- oder `partNumber`-Parameter im Ereigniskontext enthalten. Die Position des Parameters im Ereigniskontext hängt davon ab, welcher Parameter verwendet wurde und wie er in die ursprüngliche Anforderung an den Object Lambda Access Point aufgenommen wurde, wie in der folgenden Tabelle erläutert.

Parameter	Ort des Ereigniskontexts
Range (Header)	<code>userRequest.headers.Range</code>
Range (Abfrageparameter)	<code>userRequest.url</code> (Abfrageparameter <code>Range</code>)
<code>partNumber</code>	<code>userRequest.url</code> (Abfrageparameter <code>partNumber</code>)

⚠ Important

Das bereitgestellte, URL für Ihren Object Lambda Access Point vorsignierte Objekt enthält nicht den `partNumber` Parameter Range oder aus der ursprünglichen Anfrage. Sehen Sie sich die folgenden Optionen zur Handhabung dieser Parameter in Ihrem AWS Lambda Funktion.

Nachdem Sie den Range- oder `partNumber`-Wert extrahiert haben, können Sie einen der folgenden Ansätze wählen, basierend auf den Anforderungen Ihrer Anwendung:

A. Ordnen Sie den angeforderten **Range**- oder **partNumber**-Wert dem transformierten Objekt zu (empfohlen).

Die zuverlässigste Art, Range- oder `partNumber`-Anforderungen zu bearbeiten, ist folgende:

- Rufen Sie das vollständige Objekt aus Amazon S3 ab.
- Transformieren Sie das Objekt.
- Wenden Sie die angeforderten Range- oder `partNumber`-Parameter auf das transformierte Objekt an.

Verwenden Sie dazu das bereitgestellte Presigned, URL um das gesamte Objekt von Amazon S3 abzurufen und das Objekt dann nach Bedarf zu verarbeiten. Ein Beispiel für eine Lambda-Funktion, die einen Range Parameter auf diese Weise verarbeitet, finden Sie in [diesem Beispiel](#) in der AWS GitHub Beispiel-Repository.

B. Ordnen Sie das Angeforderte **Range** dem Vorsignierten URL zu.

In einigen Fällen kann Ihre Lambda-Funktion die Anfrage Range direkt der vorsignierten zuordnen, URL um nur einen Teil des Objekts aus Amazon S3 abzurufen. Dieser Ansatz ist nur dann geeignet, wenn Ihre Transformation die beiden folgenden Kriterien erfüllt:

1. Ihre Transformationsfunktion kann auf partielle Objektbereiche angewendet werden.
2. Das Anwenden der Range-Parameter vor oder nach der Transformationsfunktion führt zu demselben transformierten Objekt.

Beispielsweise erfüllt eine Transformationsfunktion, die alle Zeichen in einem ASCII -codierten Objekt in Großbuchstaben umwandelt, beide der oben genannten Kriterien. Die Transformation kann auf einen Teil eines Objekts angewendet werden, und durch Anwenden des Range-

Parameters vor der Transformation wird das gleiche Ergebnis erzielt wie beim Anwenden des Parameters nach der Transformation.

Im Gegensatz dazu erfüllt eine Funktion, die die Zeichen in einem ASCII -codierten Objekt umkehrt, diese Kriterien nicht. Eine solche Funktion erfüllt Kriterium 1, da sie auf partielle Objektbereiche angewendet werden kann. Sie erfüllt jedoch nicht Kriterium 2, da das Anwenden des Range-Parameters vor der Transformation andere Ergebnisse erzielt als das Anwenden des Parameters nach der Transformation.

Betrachten Sie eine Anforderung, die Funktion auf die ersten drei Zeichen eines Objekts mit dem Inhalt abcdefg anzuwenden. Das Anwenden des Range-Parameters vor der Transformation ruft nur abcab und kehrt dann die Daten um und gibt cba zurück. Wenn der Parameter jedoch nach der Transformation angewendet wird, ruft die Funktion das gesamte Objekt ab, kehrt es um und wendet dann den Range-Parameter, um gfe zurückzugeben. Da diese Ergebnisse unterschiedlich sind, sollte diese Funktion den Range-Parameter beim Abrufen des Objekts von Amazon S3 nicht anwenden. Stattdessen sollte es das gesamte Objekt abrufen, die Transformation durchführen und erst dann den Range-Parameter anwenden.

Warning

In vielen Fällen führt die Anwendung des Range Parameters auf den vorsignierten Client URL zu unerwartetem Verhalten der Lambda-Funktion oder des anfordernden Clients. Wenn Sie nicht sicher sind, dass Ihre Anwendung ordnungsgemäß funktioniert, wenn Sie nur ein Teilobjekt von Amazon S3 abrufen, empfehlen wir Ihnen, vollständige Objekte abzurufen und zu transformieren, wie zuvor in Ansatz A beschrieben.

Wenn Ihre Anwendung die zuvor in Ansatz B beschriebenen Kriterien erfüllt, können Sie Ihre AWS Lambda funktionieren, indem Sie nur den angeforderten Objektbereich abrufen und dann Ihre Transformation für diesen Bereich ausführen.

Im folgenden Java-Codebeispiel wird folgende Vorgehensweise gezeigt:

- Rufen Sie den Range-Header aus der GetObject-Anforderung ab.
- Fügen Sie dem vorsignierten Range Header den Header hinzuURL, den Lambda verwenden kann, um den angeforderten Bereich von Amazon S3 abzurufen.

```
private HttpRequest.Builder applyRangeHeader(ObjectLambdaEvent event,  
HttpRequest.Builder presignedRequest) {
```



```
var header = event.getUserRequest().getHeaders().entrySet().stream()
    .filter(e -> e.getKey().toLowerCase(Locale.ROOT).equals("range"))
    .findFirst();

// Add check in the query string itself.
header.ifPresent(entry -> presignedRequest.header(entry.getKey(),
entry.getValue()));
return presignedRequest;
}
```

Die Verwendung von AWS gebaute Lambda-Funktionen

AWS bietet einige vorgefertigte AWS Lambda Funktionen, die Sie mit Amazon S3 Object Lambda verwenden können, um personenbezogene Daten (PII) zu erkennen und zu redigieren und S3-Objekte zu dekomprimieren. Diese Lambda-Funktionen sind verfügbar in AWS Serverless Application Repository. Sie können diese Funktionen über den AWS Management Console wenn Sie Ihren Object Lambda Access Point erstellen.

Weitere Informationen zur Bereitstellung serverloser Anwendungen finden Sie auf AWS Serverless Application Repository, siehe [Bereitstellen von Anwendungen](#) in der AWS Serverless Application Repository Leitfaden für Entwickler.

Note

Die folgenden Beispiele können nur mit GetObject-Anforderungen verwendet werden.

Beispiel 1: PII Zugriffskontrolle

Diese Lambda-Funktion verwendet Amazon Comprehend, einen Service zur Verarbeitung natürlicher Sprache (NLP), der maschinelles Lernen nutzt, um Erkenntnisse und Zusammenhänge im Text zu finden. Diese Funktion erkennt automatisch personenbezogene Daten (PII) wie Namen, Adressen, Daten, Kreditkartennummern und Sozialversicherungsnummern in Dokumenten in Ihrem Amazon S3 S3-Bucket. Wenn Sie Dokumente in Ihrem Bucket haben, die Folgendes beinhalten PII, können Sie die PII Zugriffskontrollfunktion so konfigurieren, dass sie diese PII Entitätstypen erkennt und den Zugriff auf nicht autorisierte Benutzer einschränkt.

Stellen Sie zunächst die folgende Lambda-Funktion in Ihrem Konto bereit und fügen Sie den Amazon-Ressourcennamen (ARN) für die Funktion zu Ihrer Object Lambda Access Point-Konfiguration hinzu.

Im Folgenden finden Sie ein Beispiel ARN für diese Funktion:

```
arn:aws:serverlessrepo:us-east-1:111122223333:applications/  
ComprehendPiiAccessControlS3ObjectLambda
```

Sie können diese Funktion auf der Seite hinzufügen oder anzeigen AWS Management Console indem Sie Folgendes verwenden AWS Serverless Application Repository Verknüpfung:

[ComprehendPiiAccessControlS3 ObjectLambda](#).

Informationen zur Anzeige dieser Funktion finden Sie unter [Amazon Comprehend S3 Object Lambda](#).
GitHub

Beispiel 2: Schwärzung PII

Diese Lambda-Funktion verwendet Amazon Comprehend, einen Service zur Verarbeitung natürlicher Sprache (NLP), der maschinelles Lernen nutzt, um Erkenntnisse und Zusammenhänge im Text zu finden. Diese Funktion löscht automatisch personenbezogene Daten (PII), wie Namen, Adressen, Daten, Kreditkartennummern und Sozialversicherungsnummern, aus Dokumenten in Ihrem Amazon S3 S3-Bucket.

Wenn Sie Dokumente in Ihrem Bucket haben, die Informationen wie Kreditkartennummern oder Bankkontoinformationen enthalten, können Sie die PII Redaction S3 Object Lambda-Funktion so konfigurieren, dass sie diese Dokumente, in denen PII Entitätstypen geschwärzt sind, erkennt PII und anschließend eine Kopie zurückgibt.

Stellen Sie zunächst die folgende Lambda-Funktion in Ihrem Konto bereit und fügen Sie die ARN für die Funktion zu Ihrer Object Lambda Access Point-Konfiguration hinzu.

Im Folgenden finden Sie ein Beispiel ARN für diese Funktion:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/  
ComprehendPiiRedactionS3ObjectLambda
```

Sie können diese Funktion auf der Seite hinzufügen oder anzeigen AWS Management Console indem Sie Folgendes verwenden AWS Serverless Application Repository Verknüpfung:

[ComprehendPiiRedactionS3 ObjectLambda](#).

Informationen zur Anzeige dieser Funktion finden Sie unter [Amazon Comprehend S3 Object Lambda](#).
GitHub

Vollständige end-to-end Verfahren für einige S3 Object Lambda-Aufgaben in PII Redaction finden Sie unter. [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#)

Beispiel 3: Dekompression

Die Lambda-Funktion `S3objectLambdaDecompression` kann Objekte, die in Amazon S3 gespeichert sind, in einem von sechs komprimierten Dateiformaten dekomprimieren: `bzip2`, `gzip`, `snappy`, `zlib`, `zstandard` und `ZIP`.

Stellen Sie zunächst die folgende Lambda-Funktion in Ihrem Konto bereit und fügen Sie die ARN für die Funktion zu Ihrer Object Lambda Access Point-Konfiguration hinzu.

Im Folgenden finden Sie ein Beispiel ARN für diese Funktion:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/S3objectLambdaDecompression
```

Sie können diese Funktion auf der Seite hinzufügen oder anzeigen AWS Management Console indem Sie Folgendes verwenden AWS Serverless Application Repository Verknüpfung: [S3 ObjectLambdaDecompression](#).

Informationen zur Aktivierung dieser Funktion finden Sie GitHub unter [S3 Object Lambda Decompression](#).

Bewährte Methoden und Richtlinien für S3 Object Lambda

Befolgen Sie bei der Verwendung von S3 Object Lambda diesen bewährten Methoden und Richtlinien, um den Betrieb und die Leistung zu optimieren.

Themen

- [Arbeiten mit S3 Object Lambda](#)
- [AWS-Services wird in Verbindung mit S3 Object Lambda verwendet](#)
- [Range- und partNumber-Header](#)
- [Transformieren von expiry-date](#)
- [Ich arbeite mit dem AWS CLI and AWS SDKs](#)

Arbeiten mit S3 Object Lambda

S3 Object Lambda unterstützt nur die Verarbeitung von GET-, LIST- und HEAD-Anforderungen. Alle anderen Anfragen werden nicht aufgerufen AWS Lambda und geben stattdessen standardmäßige, nicht transformierte Antworten API zurück. Sie können maximal 1.000 Object Lambda Access Points pro AWS-Konto pro Region. Das Tool AWS Lambda Die Funktion, die Sie verwenden, muss sich in derselben befinden AWS-Konto und Region als Object Lambda Access Point.

S3 Object Lambda lässt bis zu 60 Sekunden zu, um eine vollständige Antwort zu seinen Aufrufer zu streamen. Ihre Funktion unterliegt außerdem AWS Lambda Standardkontingente. Weitere Informationen finden Sie unter [Lambda-Kontingente](#) im AWS Lambda Leitfaden für Entwickler.

Mit S3 Object Lambda wird Ihre angegebene Lambda-Funktion aufgerufen. Sie sind dafür verantwortlich, dass alle Daten, die von Ihrer angegebenen Lambda-Funktion oder -Anwendung aus S3 überschrieben oder gelöscht werden, beabsichtigt und korrekt sind.

Sie können S3 Object Lambda nur verwenden, um Operationen auf Objekten durchzuführen. Sie können S3 Object Lambda nicht verwenden, um andere Amazon-S3-Operationen auszuführen, z. B. das Ändern oder Löschen von Buckets. Eine vollständige Liste der S3-Vorgänge, die Zugriffspunkte unterstützen, finden Sie unter [Zugriffspunkt-Kompatibilität mit S3-Vorgänge](#).

Zusätzlich zu dieser Liste unterstützen Object Lambda Access Points die [SelectObjectContent](#) API Operationen [POST Object](#), [CopyObject](#) (als Quelle) und nicht.

AWS-Services wird in Verbindung mit S3 Object Lambda verwendet

S3 Object Lambda verbindet Amazon S3, AWS Lambda und optional andere AWS-Services Ihrer Wahl, um Objekte bereitzustellen, die für die anfragenden Anwendungen relevant sind. Alle AWS-Services Die mit S3 Object Lambda verwendeten Produkte unterliegen ihren jeweiligen Service Level Agreements (SLAs). Zum Beispiel, falls vorhanden AWS-Service erfüllt seine Serviceverpflichtung nicht, sind Sie berechtigt, eine Servicegutschrift zu erhalten, wie in der Servicevereinschätzung dokumentiert SLA.

Range- und partNumber-Header

Wenn Sie mit großen Objekten arbeiten, können Sie den Range HTTP Header verwenden, um einen bestimmten Bytebereich von einem Objekt herunterzuladen. Wenn Sie den Range-Header verwenden, ruft Ihre Anforderung nur den angegebenen Teil des Objekts ab. Mit dem partNumber-Header können Sie auch eine Bereichsanforderung für das angegebene Teil aus dem Objekt ausführen.

Weitere Informationen finden Sie unter [Arbeite mit Range and partNumber Header](#).

Transformieren von **expiry-date**

Sie können transformierte Objekte von Ihrem Object Lambda Access Point auf der AWS Management Console. Diese Objekte dürfen nicht abgelaufen sein. Wenn Ihre Lambda-Funktion das `expiry-date` Ihrer Objekte transformiert, sehen Sie möglicherweise abgelaufene Objekte, die nicht geöffnet oder heruntergeladen werden können. Dieses Verhalten gilt nur für wiederhergestellte Objekte von S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive.

Ich arbeite mit dem AWS CLI and AWS SDKs

AWS Command Line Interface (AWS CLI) S3-Unterbefehle (`cpmv`, `undsync`) und die Verwendung von AWS SDK for Java `TransferManager` Klassen werden für die Verwendung mit S3 Object Lambda nicht unterstützt.

Tutorials zu S3 Object Lambda

In den folgenden Tutorials end-to-end werden vollständige Verfahren für einige S3 Object Lambda-Aufgaben vorgestellt.

- [Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#)
- [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#)
- [Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen](#)

Debuggen von S3 Object Lambda

Anforderungen an Zugriffspunkte von Amazon S3 Object Lambda können zu neuen Fehlerantworten führen, wenn mit dem Aufruf oder der Ausführung der Lambda-Funktion etwas schief geht. Diese Fehler folgen dem gleichen Format wie die Standardfehler von Amazon S3. Informationen zu S3 Object Lambda-Fehlern finden Sie in der [Liste der S3 Object Lambda-Fehlercodes](#) in der Amazon Simple Storage Service API Reference.

Weitere Informationen zum allgemeinen Debuggen von Lambda-Funktionen finden Sie unter [Überwachung und Problembehandlung von Lambda-Anwendungen](#) in der AWS Lambda Leitfaden für Entwickler.

Informationen zu standardmäßigen Amazon S3 S3-Fehlern finden Sie unter [Error Responses](#) in der Amazon Simple Storage Service API Reference.

Sie können in Amazon Anforderungsmetriken CloudWatch für Ihre Object Lambda Access Points aktivieren. Diese Metriken helfen Ihnen dabei, die Betriebsleistung Ihrer Zugriffspunkte zu überwachen. Sie können Anforderungsmetriken während oder nach der Erstellung Ihres Object Lambda Access Point aktivieren. Weitere Informationen finden Sie unter [Lambda-Anforderungsmetriken für S3-Objekte in CloudWatch](#).

Um eine detailliertere Protokollierung von Anfragen an Ihre Object Lambda Access Points zu erhalten, können Sie Folgendes aktivieren: AWS CloudTrail Datenereignisse. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen für Pfade](#) in der AWS CloudTrail Benutzerleitfaden.

Informationen zu S3-Objekt-Lambda-Tutorials finden Sie im Folgenden:

- [Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda](#)
- [Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend](#)
- [Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen](#)

Weitere Hinweise zu standardmäßigen Zugriffspunkten finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Hinweise zum Arbeiten mit Buckets finden Sie unter [Bucket-Übersicht](#). Weitere Informationen zur Arbeit mit Objekten finden Sie unter [Übersicht über Amazon-S3-Objekte](#).

Was ist S3 Express One Zone?

Amazon S3 Express One Zone ist eine leistungsstarke Amazon-S3-Speicherklasse mit einer einzelnen Zone, die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die derzeit verfügbare Cloud-Objektspeicherklasse mit der niedrigsten Latenz. Sie bietet Datenzugriffsgeschwindigkeiten, die bis zu zehnmal schneller sind und die Anforderungskosten um 50 Prozent niedriger sind als bei S3 Standard. Anwendungen können sofort davon profitieren, dass Anfragen bis zu einer Größenordnung schneller bearbeitet werden. S3 Express One Zone bietet eine ähnliche Leistungselastizität wie andere S3-Speicherklassen.

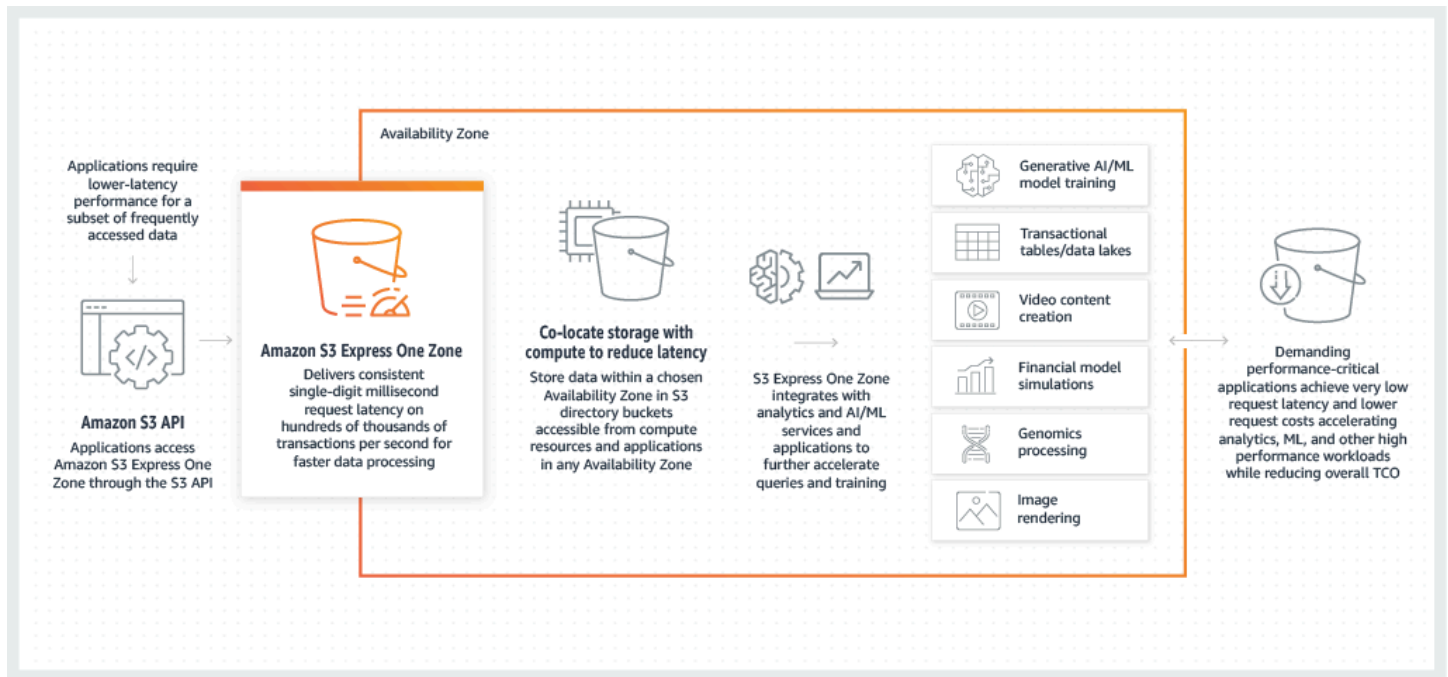
Wie bei anderen Amazon S3 S3-Speicherklassen müssen Sie die Kapazitäts- oder Durchsatzanforderungen nicht im Voraus planen oder bereitstellen. Sie können Ihren Speicher je nach Bedarf nach oben oder unten skalieren und über Amazon S3 auf Ihre Daten zugreifenAPI.

S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Um die Zugriffsgeschwindigkeit weiter zu erhöhen und Hunderttausende von Anfragen pro Sekunde zu unterstützen, werden Daten der Speicherklasse S3 Express One Zone zusätzlich in einem neuen Bucket-Typ gespeichert: einem Amazon S3-Verzeichnis-Bucket. Jeder Verzeichnis-Bucket kann Hunderttausende von Transaktionen pro Sekunde (TPS) unterstützen, unabhängig von Schlüsselnamen oder Zugriffsmustern.

Die Speicherklasse Amazon S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das [Amazon S3 Service Level Agreement](#) unterstützt. Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone wurde entwickelt, um mit gleichzeitigen Geräteausfällen umzugehen, indem verlorene Redundanz schnell erkannt und behoben werden kann. Wenn das vorhandene Gerät ausfällt, leitet S3 Express One Zone Anfragen automatisch an neue Geräte innerhalb einer Availability Zone weiter. Diese Redundanz trägt dazu bei, den unterbrechungsfreien Zugriff auf Ihre Daten innerhalb einer Availability Zone sicherzustellen.

S3 Express One Zone ist ideal für jede Anwendung geeignet, bei der es wichtig ist, die für den Zugriff auf ein Objekt erforderliche Latenz zu minimieren. Bei solchen Anwendungen kann es sich um interaktive Workflows handeln, z. B. bei der Videobearbeitung, bei der Kreative über ihre Benutzeroberflächen reaktionsschnellen Zugriff auf Inhalte benötigen. S3 Express One Zone bietet auch Vorteile für Analysen und maschinelles Lernen, die ähnliche Anforderungen an die

Reaktionsfähigkeit ihrer Daten stellen, insbesondere Workloads mit vielen kleineren Zugriffen oder einer großen Anzahl zufälliger Zugriffe. S3 Express One Zone kann mit anderen verwendet werden AWS-Services zur Unterstützung von Workloads in den Bereichen Analytik, künstliche Intelligenz und maschinelles Lernen (KI/ML) wie AmazonEMR, Amazon und Amazon SageMaker Athena.



Wenn Sie S3 Express One Zone verwenden, können Sie mithilfe eines VPC Gateway-Endpunkts mit Ihrem Verzeichnis-Bucket in einer virtuellen privaten Cloud (VPC) interagieren. Mit einem Gateway-Endpunkt können Sie von Ihrem Computer aus auf S3 Express One Zone-Verzeichnis-Buckets zugreifen, VPC ohne dass Sie ein Internet-Gateway oder ein NAT Gerät VPC benötigen, und das ohne zusätzliche Kosten.

Sie können viele der gleichen Amazon S3 API S3-Operationen und -Funktionen mit Verzeichnis-Buckets verwenden, die Sie mit Allzweck-Buckets und anderen Speicherklassen verwenden. Dazu gehören Mountpoint für Amazon S3, serverseitige Verschlüsselung mit Amazon S3 S3-verwalteten Schlüsseln (SSE-S3), serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Tasten (SSE-KMS), S3 Batch Operations und S3 Block Public Access. Sie können über die Amazon S3-Konsole auf S3 Express One Zone zugreifen. AWS Command Line Interface (AWS CLI), AWS SDKs und der Amazon S3 RESTAPI.

Weitere Informationen zu S3 Express One Zone finden Sie in den folgenden Themen.

- [Übersicht](#)
- [Funktionen von S3 Express One Zone](#)

- [Zugehörige Services](#)
- [Nächste Schritte](#)

Übersicht

Um die Leistung zu optimieren und die Latenz zu reduzieren, führt S3 Express One Zone die folgenden neuen Konzepte ein.

Einzelne Availability Zone

Die Speicherklasse Amazon S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das [Amazon S3 Service Level Agreement](#) unterstützt. Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone wurde entwickelt, um mit gleichzeitigen Geräteausfällen umzugehen, indem verlorene Redundanz schnell erkannt und behoben werden kann. Wenn das vorhandene Gerät ausfällt, leitet S3 Express One Zone Anfragen automatisch an neue Geräte innerhalb einer Availability Zone weiter. Diese Redundanz trägt dazu bei, den unterbrechungsfreien Zugriff auf Ihre Daten innerhalb einer Availability Zone sicherzustellen.

Eine Availability Zone besteht aus einem oder mehreren diskreten Rechenzentren mit redundanter Stromversorgung, Vernetzung und Konnektivität in einem AWS-Region. Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie die Availability Zone und AWS-Region wo sich Ihr Bucket befinden wird.

Verzeichnis-Buckets

Es gibt zwei Arten von Amazon S3 S3-Buckets: S3-Allzweck-Buckets und S3-Verzeichnis-Buckets. Allzweck-Buckets sind der standardmäßige Amazon-S3-Bucket-Typ, der für die überwiegende Mehrheit der S3-Anwendungsfälle verwendet wird. Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone, die für Workloads oder leistungskritische Anwendungen konzipiert ist, die eine konsistente Latenz im einstelligen Millisekundenbereich erfordern. Wählen Sie den Bucket-Typ, der am besten zu Ihren Anwendungs- und Leistungsanforderungen passt.

Verzeichnis-Buckets organisieren Daten hierarchisch in Verzeichnissen, im Gegensatz zu der flachen Speicherstruktur von Allzweck-Buckets. Es gibt keine Präfixbeschränkungen für Verzeichnis-Buckets und einzelne Verzeichnisse können horizontal skaliert werden.

Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone, die für die Verwendung durch leistungssensitive Anwendungen konzipiert wurde. Mit S3 Express One Zone können Sie

eine einzelne Availability Zone mit der Option auswählen, Ihren Objektspeicher gemeinsam mit Ihren Rechenressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Dies unterscheidet sich von Allzweck-Buckets, in denen Objekte redundant in mehreren Availability Zones gespeichert werden AWS-Regionen.

Weitere Information zu Verzeichnis-Buckets finden Sie unter [Verzeichnis-Buckets](#). Weitere Informationen über Allzweck-Buckets finden Sie unter [Bucket-Übersicht](#).

Endpunkte und Gateway-Endpunkte VPC

APIBucket-Management-Operationen für Directory-Buckets sind über einen regionalen Endpunkt verfügbar und werden als regionale Endpunktoperationen bezeichnet. API Beispiele für regionale API Endpunktoperationen sind `CreateBucket` `DeleteBucket`. Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie Zonal Endpoint API Operations verwenden, um die Objekte in Ihrem Verzeichnis-Bucket hochzuladen und zu verwalten. Zonale API Endpunktoperationen sind über einen zonalen Endpunkt verfügbar. Beispiele für API Operationen an zonalen Endpunkten sind `PutObject` `CopyObject`.

Sie können über VPC Gateway-Endpunkte VPC von Ihrer aus auf S3 Express One Zone zugreifen. Nachdem Sie einen Gateway-Endpunkt erstellt haben, können Sie ihn als Ziel in Ihrer Routentabelle für den Verkehr hinzufügen, der von Ihrer VPC zur S3 Express One Zone bestimmt ist. Wie bei Amazon S3 fallen für die Nutzung von Gateway-Endpunkten keine zusätzlichen Gebühren an. Weitere Informationen zur Konfiguration von VPC Gateway-Endpunkten finden Sie unter [Networking für S3 Express One Zone](#).

Sitzungsbasierte Autorisierung

Mit S3 Express One Zone authentifizieren und autorisieren Sie Anfragen über einen neuen sitzungsbasierten Mechanismus, der für die geringste Latenz optimiert ist. Sie können mit `CreateSession` temporäre Anmeldeinformationen anfordern, die den Zugriff auf Ihren Bucket mit geringer Latenz ermöglichen. Diese temporären Anmeldeinformationen sind einem bestimmten S3-Verzeichnis-Bucket zugeordnet. Sitzungstoken werden nur bei zonalen Vorgängen (auf Objektebene) verwendet (mit Ausnahme von [CopyObject](#)). Weitere Informationen finden Sie unter [CreateSession-Autorisierung](#).

Die [unterstützten AWS SDKs für S3 Express One Zone](#) kümmern Sie sich in Ihrem Namen um den Aufbau und die Aktualisierung von Sitzungen. Um Ihre Sitzungen zu schützen, laufen temporäre Sicherheitsanmeldedaten nach 5 Minuten ab. Nachdem Sie das heruntergeladen und installiert

haben AWS SDKs und konfigurieren Sie das Notwendige AWS Identity and Access Management (IAM) Berechtigungen, Sie können sofort mit der Nutzung von API Operationen beginnen.

Funktionen von S3 Express One Zone

Die folgenden S3-Funktionen sind für S3 Express One Zone verfügbar. Eine vollständige Liste der unterstützten API Operationen und nicht unterstützten Funktionen finden Sie unter [Wodurch zeichnet sich S3 Express One Zone aus?](#)

Zugriffsverwaltung und Sicherheit

Mit Verzeichnis-Buckets können Sie die folgenden Funktionen zum Überwachen und Verwalten des Zugriffs verwenden. Standardmäßig sind Verzeichnis-Buckets privat und sind nur für Benutzer zugänglich, denen explizit Zugriff gewährt wurde. Im Gegensatz zu Allzweck-Buckets, bei denen die Zugriffskontrollgrenze auf Bucket-, Präfix- oder Objekt-Tag-Ebene festgelegt werden kann, wird die Zugriffskontrollgrenze für Verzeichnis-Buckets nur auf Bucket-Ebene festgelegt. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

- [S3 Block Public Access](#) — Alle S3 Block Public Access-Einstellungen sind standardmäßig auf Bucket-Ebene aktiviert. Diese Standardeinstellung kann nicht geändert werden.
- [S3 Object Ownership](#) (Bucket-Besitzer wird standardmäßig erzwungen) — Zugriffskontrolllisten (ACLs) werden für Verzeichnis-Buckets nicht unterstützt. Directory-Buckets verwenden automatisch die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum. Bucket Owner erzwungen bedeutet, dass diese deaktiviert ACLs sind. Der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle über jedes Objekt im Bucket. Diese Standardeinstellung kann nicht geändert werden.
- [AWS Identity and Access Management \(IAM\)](#) — IAM hilft Ihnen dabei, den Zugriff auf Ihre Verzeichnis-Buckets sicher zu kontrollieren. Mit dieser Aktion IAM können Sie Zugriff auf API Bucket-Management-Operationen (Regional) und API Objektmanagement-Operationen (Zonal) gewähren. `s3express:CreateSession` Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#). Im Gegensatz zu Objektverwaltungsaktionen können Bucket-Verwaltungsaktionen nicht kontoübergreifend sein. Nur der Bucket-Eigentümer kann diese Aktionen ausführen.
- [Bucket-Richtlinien](#) — Verwenden Sie eine IAM basierte Richtlinienprache, um ressourcenbasierte Berechtigungen für Ihre Verzeichnis-Buckets zu konfigurieren. Sie können sie auch verwenden IAM,

um den Zugriff auf den `CreateSession` API Vorgang zu steuern, sodass Sie die zonalen Operationen oder die Objektverwaltung verwenden können. API Sie können über dasselbe Konto oder kontoübergreifenden Zugriff auf zonale Operationen gewähren. API Weitere Informationen zu den Berechtigungen und Richtlinien für S3 Express One Zone finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#)

- [IAM Access Analyzer for S3](#) — Evaluieren und überwachen Sie Ihre Zugriffsrichtlinien, um sicherzustellen, dass die Richtlinien nur den beabsichtigten Zugriff auf Ihre S3-Ressourcen ermöglichen.

Protokollierung und Überwachung

S3 Express One Zone verwendet die folgenden S3-Protokollierungs- und Überwachungstools, mit denen Sie die Nutzung Ihrer Ressourcen überwachen und kontrollieren können:

- [CloudWatch Amazon-Metriken](#) — Überwachen Sie Ihre AWS Ressourcen und Anwendungen, die CloudWatch zur Erfassung und Nachverfolgung von Metriken verwendet werden. S3 Express One Zone verwendet denselben CloudWatch Namespace wie andere Amazon S3 S3-Speicherklassen (AWS/S3) und unterstützt tägliche Speichermetriken für Verzeichnis-Buckets: `BucketSizeBytes` und `NumberOfObjects` Weitere Informationen finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).
- [AWS CloudTrail Protokolle](#) — AWS CloudTrail ist ein AWS-Service das Sie bei der Durchführung von Betriebs- und Risikoprüfungen, Unternehmensführung und Einhaltung Ihrer AWS-Konto indem es die Aktionen aufzeichnet, die von einem Benutzer, einer Rolle oder einem AWS-Service. CloudTrail Erfasst für S3 Express One Zone regionale API Endpunktoperationen (zum Beispiel `CreateBucket` und `PutBucketPolicy`) als Verwaltungsereignisse und zonale API Operationen (zum Beispiel `GetObject` und `PutObject`) als Datenereignisse. Zu diesen Ereignissen gehören Maßnahmen im AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDKs, und AWS API Operationen. Weitere Informationen finden Sie unter [Protokollierung mit AWS CloudTrail für S3 Express One Zone](#).

Note

Amazon S3 S3-Serverzugriffsprotokolle werden mit S3 Express One Zone nicht unterstützt.

Verwaltung von Objekten

Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie Ihren Objektspeicher mithilfe der Amazon S3 S3-Konsole verwalten. AWS SDKs, und AWS CLI. Die folgenden Funktionen sind für die Objektverwaltung mit S3 Express One Zone verfügbar:

- [S3 Batch Operations](#) — Verwenden Sie Batch-Operationen, um Massenoperationen für Objekte in Verzeichnis-Buckets durchzuführen, z. B. Kopieren und Aufrufen AWS Lambda Funktion. Sie können mit Batch Operations beispielsweise Objekte zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Mit Batch Operations können Sie Milliarden von Objekten in großem Umfang mit einer einzigen S3-Anfrage verwalten, indem Sie den AWS SDKs oder AWS CLI oder ein paar Klicks in der Amazon S3 S3-Konsole.
- [Import](#) – Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie Ihren Bucket mithilfe der Importfunktion in der Amazon-S3-Konsole mit Objekten füllen. Import ist eine optimierte Methode zur Erstellung von Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets.

AWS SDKs und Client-Bibliotheken

Nachdem Sie einen Verzeichnis-Bucket erstellt und ein Objekt in Ihren Bucket hochgeladen haben, können Sie Ihren Objektspeicher wie folgt verwalten.

- [Mountpoint for Amazon S3](#) — Mountpoint for Amazon S3 ist ein Open-Source-Dateiclient, der Zugriff mit hohem Durchsatz ermöglicht und so die Rechenkosten für Data Lakes auf Amazon S3 senkt. Mountpoint für Amazon S3 übersetzt lokale Dateisystem-Aufrufe in API S3-Objekt-Aufrufe wie `GET LIST`. Es ist ideal für leseintensive Data Lake-Workloads, die Petabyte an Daten verarbeiten und den von Amazon S3 bereitgestellten hohen elastischen Durchsatz benötigen, um über Tausende von Instances nach oben und unten zu skalieren.
- [S3A](#) – S3A ist ein empfohlener Hadoop-kompatible Schnittstelle für den Zugriff auf Datenspeicher in Amazon S3. S3A ersetzt die S3N Hadoop Dateisystem-Client.
- [PyTorch on AWS](#) — PyTorch ein AWS ist ein Open-Source-Deep-Learning-Framework, das es einfacher macht, Modelle für maschinelles Lernen zu entwickeln und in der Produktion einzusetzen.
- [AWS SDKs](#) — Sie können das verwenden AWS SDKs bei der Entwicklung von Anwendungen mit Amazon S3. Das Tool AWS SDKs vereinfachen Sie Ihre Programmieraufgaben, indem Sie das zugrunde liegende Amazon S3 einschließen RESTAPI. Weitere Informationen zur Verwendung von AWS SDKs mit S3 Express One Zone finden Sie unter [the section called “AWS SDKs”](#).

Verschlüsselung und Datenschutz

In Verzeichnis-Buckets gespeicherte Objekte werden automatisch mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Verzeichnis-Buckets unterstützen auch serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS). Directory-Buckets unterstützen keine serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) oder serverseitige Dual-Layer-Verschlüsselung mit AWS KMS keys (-). DSSE KMS Weitere Informationen finden Sie unter [Datenschutz und Verschlüsselung in S3 Express One Zone](#).

S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) wählen: CRC32, CRC32C, SHA -1 und -256. SHA MD5basierte Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

Weitere Informationen finden Sie unter [Bewährte Methoden für zusätzliche S3-Prüfsummen](#).

AWS Signatur Version 4 (SigV4)

S3 Express One Zone verwendet AWS Signatur Version 4 (SigV4). SigV4 ist ein Signaturprotokoll, das zur Authentifizierung von Anfragen an Amazon S3 verwendet wird. HTTPS S3 Express One Zone signiert Anfragen mithilfe von AWS Sigv4. Weitere Informationen finden Sie unter [Authentifizieren von Anfragen \(AWS Signature Version 4\)](#) in der Amazon Simple Storage API Service-Referenz.

Starke Konsistenz

S3 Express One Zone bietet eine hohe read-after-write Konsistenz für PUT alle Objekte und DELETE Anfragen von Objekten in Ihren Verzeichnis-Buckets AWS-Regionen. Weitere Informationen finden Sie unter [Amazon S3-Datenkonsistenzmodell](#).

Zugehörige Services

Sie können Folgendes verwenden AWS-Services mit der Speicherklasse S3 Express One Zone zur Unterstützung Ihres spezifischen Anwendungsfalls mit niedriger Latenz.

- [Amazon Elastic Compute Cloud \(AmazonEC2\)](#) — Amazon EC2 bietet sichere und skalierbare Rechenkapazität in der AWS Cloud. Durch die Verwendung von Amazon EC2 müssen Sie nicht im

Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwenden EC2, um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten.

- [AWS Lambda](#) – Lambda ist ein Computing-Service, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Sie konfigurieren Benachrichtigungseinstellungen für einen Bucket und erteilen die Amazon-S3-Berechtigung zum Aufrufen einer Funktion in der ressourcenbasierten Berechtigungsrichtlinie der Funktion.
- [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) — Amazon EKS ist ein verwalteter Service, der die Installation, den Betrieb und die Wartung Ihres eigenen Services überflüssig macht. Kubernetes Kontrollebene ein AWS. [Kubernetes](#) ist ein Open-Source-System, das die Verwaltung, Skalierung und Bereitstellung von containerisierten Anwendungen automatisiert.
- [Amazon Elastic Container Service \(Amazon ECS\)](#) — Amazon ECS ist ein vollständig verwalteter Container-Orchestrierungs-Service, mit dem Sie containerisierte Anwendungen einfach bereitstellen, verwalten und skalieren können.
- [Amazon Athena](#) — Athena ist ein interaktiver Abfrageservice, der es einfach macht, Daten mithilfe von Standards direkt in Amazon S3 zu analysieren. [SQL](#) Sie können Athena auch verwenden, um Datenanalysen interaktiv auszuführen, indem Sie Apache Spark ohne Ressourcen planen, konfigurieren oder verwalten zu müssen. Wenn du rennst Apache Spark Bewerbungen auf Athena reichen Sie ein Spark Code zur Verarbeitung und direkt die Ergebnisse erhalten.
- [Amazon SageMaker Runtime Model Training](#) — Amazon SageMaker Runtime ist ein vollständig verwalteter Service für maschinelles Lernen. Mit SageMaker Runtime können Datenwissenschaftler und Entwickler schnell und einfach Modelle für maschinelles Lernen erstellen und trainieren und sie dann direkt in einer produktionsbereiten gehosteten Umgebung bereitstellen.
- [AWS Glue](#) – AWS Glue ist ein serverloser Datenintegrationsdienst, der es Analytics-Benutzern leicht macht, Daten aus mehreren Quellen zu entdecken, aufzubereiten, zu verschieben und zu integrieren. Sie können Folgendes verwenden ... AWS Glue für Analytik, maschinelles Lernen und Anwendungsentwicklung. AWS Glue umfasst auch zusätzliche Produktivitäts- und Datenmanagement-Tools für die Erstellung, Ausführung von Jobs und Implementierung von Geschäftsabläufen.
- [Amazon EMR](#) — Amazon EMR ist eine verwaltete Cluster-Plattform, die den Betrieb von Big-Data-Frameworks vereinfacht, wie Apache Hadoop and Apache Spark, auf AWS um riesige Datenmengen zu verarbeiten und zu analysieren.

Nächste Schritte

Weitere Informationen zum Arbeiten mit der Speicherklasse S3 Express One Zone und Verzeichnis-Buckets finden Sie in den folgenden Themen:

- [Wodurch zeichnet sich S3 Express One Zone aus?](#)
- [Tutorial: Erste Schritte mit S3 Express One Zone](#)
- [Networking für S3 Express One Zone](#)
- [Verzeichnis-Buckets](#)
- [Arbeiten mit Objekten in einem Verzeichnis-Bucket](#)
- [Sicherheit für Verzeichnis-Buckets](#)
- [Optimieren der Leistung von Amazon S3 Express One Zone](#)
- [Entwicklung mit S3 Express One Zone](#)

Wodurch zeichnet sich S3 Express One Zone aus?

Amazon S3 Express One Zone ist eine leistungsstarke Amazon-S3-Speicherklasse mit einer einzelnen Zone, die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Um die Zugriffsgeschwindigkeit weiter zu erhöhen und Hunderttausende von Anfragen pro Sekunde zu unterstützen, werden S3-Express-One-Zone-Daten außerdem in einem neuen Bucket-Typ gespeichert: einem Amazon-S3-Verzeichnis-Bucket.

Weitere Informationen erhalten Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Mit Amazon S3 können Sie Verzeichnis-Buckets erstellen und auf Ihre Daten in S3 Express One Zone zugreifen. API Amazon S3 API ist mit S3 Express One Zone und Directory-Buckets kompatibel, mit Ausnahme einiger bemerkenswerter Unterschiede. Weitere Informationen zu den Unterschieden von S3 Express One Zone finden Sie in den folgenden Themen.

Themen

- [Unterschiede von S3 Express One Zone](#)
- [API von S3 Express One Zone unterstützte Operationen](#)

- [Amazon-S3-Features, die von S3 Express One Zone nicht unterstützt werden](#)

Unterschiede von S3 Express One Zone

- Unterstützter Bucket-Typ – Objekte in Verzeichnis-Buckets werden in der Speicherklasse S3 Express One Zone gespeichert. Weitere Informationen finden Sie unter [Verzeichnis-Buckets](#).
- Haltbarkeit – Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone ist für eine Verfügbarkeit von 99,95 % innerhalb einer einzigen Availability Zone konzipiert und wird durch das [Amazon S3 Service Level Agreement](#) unterstützt. Weitere Informationen finden Sie unter [Einzelne Availability Zone](#).
- **ListObjectsV2** Verhalten
 - Gibt bei Verzeichnis-Buckets ListObjectsV2 keine Objekte in lexikografischer (alphabetischer) Reihenfolge zurück. Außerdem müssen Präfixe mit einem Trennzeichen enden und es kann nur „/“ als Trennzeichen angegeben werden.
 - Bei Verzeichnis-Buckets enthält die ListObjectsV2 Antwort die Präfixe, die sich nur auf laufende mehrteilige Uploads beziehen.
- Löschverhalten – Wenn Sie ein Objekt in einem Verzeichnis-Bucket löschen, löscht Amazon S3 rekursiv alle leeren Verzeichnisse im Objektpfad. Wenn Sie beispielsweise den Objektschlüssel löschendir1/dir2/file1.txt, löscht file1.txt Amazon S3. Wenn die Verzeichnisse dir1/ und dir2/ leer sind und keine anderen Objekte enthalten, löscht Amazon S3 auch diese Verzeichnisse.
- ETags und Prüfsummen — Entitäts-Tags (ETags) für S3 Express One Zone sind zufällige alphanumerische Zeichenketten und keine Prüfsummen. MD5 Weitere Informationen zur Verwendung zusätzlicher Prüfsummen mit S3 Express One Zone finden Sie unter [Bewährte Methoden für zusätzliche S3-Prüfsummen](#).
- Objektschlüssel in **DeleteObjects**-Anforderungen
 - Objektschlüssel in DeleteObjects-Anforderungen müssen mindestens ein Zeichen enthalten, das kein Leerzeichen ist. Zeichenfolgen, die ausschließlich Leerzeichen enthalten, werden in DeleteObjects-Anforderungen nicht unterstützt.
 - Objektschlüssel in DeleteObjects-Anforderungen dürfen keine Unicode-Steuerzeichen enthalten, mit Ausnahme von newline (\n), tab (\t) und carriage return (\r).
- Regionale und zonale Endpunkte – Wenn Sie S3 Express One Zone verwenden, müssen Sie die Region in allen Client-Anfragen angeben. Für regionale Endpunkte geben Sie die Region

an, zum Beispiel `s3express-control.us-west-2.amazonaws.com`. Für zonale Endpunkte geben Sie sowohl die Region als auch die Availability Zone an, zum Beispiel `s3express-usw2-az1.us-west-2.amazonaws.com`. Weitere Informationen finden Sie unter [Regionale und zonale Endpunkte](#).

- Mehrteilige Uploads – Wie bei anderen in Amazon S3 gespeicherten Objekten können Sie große Objekte, die in der Speicherklasse S3 Express One Zone gespeichert sind, mithilfe des mehrteiligen Upload-Prozesses hochladen und kopieren. Im Folgenden sind jedoch einige Unterschiede aufgeführt, wenn Sie den mehrteiligen Upload-Prozess für Objekte verwenden, die in S3 Express One Zone gespeichert sind. Weitere Informationen finden Sie unter [the section called “Mehrteilige Uploads mit Verzeichnis-Buckets verwenden”](#).
 - Das Objekterstellungsdatum ist das Abschlussdatum des mehrteiligen Uploads.
 - Bei mehrteiligen Uploads müssen aufeinanderfolgende Teilenummern verwendet werden. Wenn Sie versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen HTTP 400 (Bad Request) Fehler.
 - Der Initiator eines mehrteiligen Uploads kann die Anforderung für den mehrteiligen Upload nur abbrechen, wenn ihm durch die entsprechende `s3express:CreateSession`-Berechtigung der Zugriff auf `AbortMultipartUpload` ausdrücklich gewährt wurde. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).
- Leeren eines Verzeichnis-Buckets — Der Befehl über `s3` im AWS Command Line Interface (CLI), die `delete` Operation über Mountpoint und die Optionsschaltfläche Bucket leeren über AWS Management Console sind nicht in der Lage, laufende mehrteilige Uploads in einem Verzeichnis-Bucket zu löschen. Um diese laufenden mehrteiligen Uploads zu löschen, führen Sie den `ListMultipartUploads` Vorgang aus, um die laufenden mehrteiligen Uploads im Bucket aufzulisten und mithilfe des Vorgangs alle laufenden mehrteiligen Uploads abzubrechen. `AbortMultipartUpload`
- Server-side encryption
 - Um Ihre Daten für Directory-Buckets mit serverseitiger Verschlüsselung zu verschlüsseln, können Sie entweder serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) (Standard) oder serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (-). SSE KMS Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschten Verschlüsselungskonfigurationen verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanfragen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum

Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).

- Ihre KMS Konfiguration kann für die gesamte Lebensdauer des Buckets nur einen vom [Kunden verwalteten Schlüssel](#) pro SSE Verzeichnis-Bucket unterstützen. [Von AWS verwalteter Schlüssel](#)(aws/s3) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben haben KMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

- Sie stellen eine HeadObject API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

- Bei [API Vorgängen an zonalen Endpunkten \(Objektebene\)](#) mit Ausnahme von und authentifizieren [CopyObject](#) und [UploadPartCopy](#) autorisieren Sie Anfragen über [CreateSession](#) für niedrige Latenz. Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanfragen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).
- Um neue Objektkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [CopyObject](#) Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objektkopien mit SSE - angeben, müssen Sie sicherstellen KMS, dass der

Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Directory-Buckets angegeben haben. Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [UploadPartCopy](#) Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den [UploadPartCopy](#)Anforderungsheadern angeben. Außerdem müssen die Verschlüsselungseinstellungen, die Sie in der [CreateMultipartUpload](#)Anfrage angeben, mit der Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.

- S3-Bucket-Keys sind immer für GET und für PUT Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE [CopyObject](#), [UploadPartCopy](#), der [Copy Operation in Batch-Operationen](#) oder [die import Jobs](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS - verschlüsseltes Objekt gestellt wird.
- Wenn Sie eine angeben [AWS KMS Verwenden Sie](#) für die Verschlüsselung in Ihrem Verzeichnis-Bucket vom Kunden verwalteten Schlüssel nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.

API von S3 Express One Zone unterstützte Operationen

Die Speicherklasse Amazon S3 Express One Zone unterstützt sowohl regionale (Bucket-Ebene oder Kontrollebene) als auch zonale (Objektebene oder Datenebene) API Endpunktoperationen. Weitere Informationen erhalten Sie unter [Networking für S3 Express One Zone](#) und [Endpunkte und Gateway-Endpunkte VPC](#).

Regionale Endpunktoperationen API

Die folgenden regionalen API Endpunktoperationen werden für S3 Express One Zone unterstützt:

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [GetBucketPolicy](#)
- [ListDirectoryBuckets](#)

- [PutBucketPolicy](#)

Zonale Endpunktoperationen API

Die folgenden zonalen API Endpunktoperationen werden für S3 Express One Zone unterstützt:

- [CreateSession](#)
- [CopyObject](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [GetObject](#)
- [GetObjectAttributes](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListObjectsV2](#)
- [PutObject](#)
- [AbortMultipartUpload](#)
- [CompleteMultiPartUpload](#)
- [CreateMultipartUpload](#)
- [ListMultipartUploads](#)
- [ListParts](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [UploadPartCopy](#)
- [PutBucketEncryption](#)
- [GetBucketEncryption](#)
- [DeleteBucketEncryption](#)

Amazon-S3-Features, die von S3 Express One Zone nicht unterstützt werden

Die folgenden Amazon-S3-Features werden von S3 Express One Zone nicht unterstützt:

- AWS Verwaltete Richtlinien
- AWS PrivateLink für S3
- MD5Prüfsummen
- Multi-Faktor-Authentifizierung () löschen MFA
- S3-Objektsperre
- Zahlung durch den Anforderer
- S3 Access Grants
- S3-Zugriffspunkte
- Bucket-Tags
- CloudWatch Amazon-Anforderungskennzahlen
- S3-Ereignis-Benachrichtigungen
- S3-Lebenszyklus
- Multiregionale S3-Zugriffspunkte
- S3-Objekt-Lambda-Zugangspunkte
- S3-Versioning
- S3-Bestand
- S3-Replikation
- Objekt-Tags
- S3 Select
- Serverzugriffsprotokolle
- Statisches Website-Hosting
- S3 Storage Lens
- S3-Storage-Lens-Gruppen
- S3 Transfer Acceleration
- Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (-) DSSE KMS
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE
- Die Option, die Einstellungen eines vorhandenen Buckets zu kopieren, wenn ein neuer Bucket in der Amazon S3 S3-Konsole erstellt wird
- Verbesserte Fehlermeldungen „Zugriff verweigert“ (HTTP403 Forbidden)

Tutorial: Erste Schritte mit S3 Express One Zone

Amazon S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher zusammen mit Ihren Rechenressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Daten in S3 Express One Zone werden in S3-Verzeichnis-Buckets gespeichert. Weitere Informationen zu Verzeichnis-Buckets finden Sie unter [Directory-Buckets](#).

S3 Express One Zone ist ideal für alle Anwendungen, bei denen es wichtig ist, die Latenz von Anfragen zu minimieren. Bei solchen Anwendungen kann es sich um Workflows handeln, bei denen Menschen interaktiv sind, z. B. bei der Videobearbeitung, bei denen Kreativprofis einen reaktionsschnellen Zugriff auf Inhalte über ihre Benutzeroberflächen benötigen. S3 Express One Zone bietet auch Vorteile für Analysen und maschinelles Lernen, die ähnliche Anforderungen an die Reaktionsfähigkeit ihrer Daten stellen, insbesondere Workloads mit vielen kleineren Zugriffen oder einer großen Anzahl zufälliger Zugriffe. S3 Express One Zone kann zusammen mit anderen verwendet werden AWS Dienste wie AmazonEMR, Amazon Athena, AWS Glue Data Catalog und Amazon SageMaker Model Training zur Unterstützung von Workloads in den Bereichen Analytik, künstliche Intelligenz und maschinelles Lernen (KI/ML),. Sie können mit der Speicherklasse und den Verzeichnis-Buckets von S3 Express One Zone arbeiten, indem Sie die Amazon S3 S3-Konsole verwenden. AWS SDKs, AWS Befehlszeilenschnittstelle (AWS CLI) und Amazon S3 RESTAPI. Weitere Informationen finden Sie unter [Was ist S3 Express One Zone?](#) und [Wie unterscheidet sich S3 Express One Zone?](#) .

Dies ist ein Workflow-Diagramm für S3 Express One Zone.

Ziel

In diesem Tutorial erfahren Sie, wie Sie einen Gateway-Endpunkt erstellen, eine IAM Richtlinie erstellen und anhängen, einen Verzeichnis-Bucket erstellen und dann mithilfe der Import-Aktion Ihren Verzeichnis-Bucket mit Objekten füllen, die derzeit in Ihrem Allzweck-Bucket gespeichert sind. Alternativ können Sie Objekte manuell in Ihren Verzeichnis-Bucket hochladen.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Konfigurieren Sie einen VPC Gateway-Endpunkt](#)
- [Schritt 2: Erstellen Sie einen Verzeichnis-Bucket](#)
- [Schritt 3: Daten in einen Verzeichnis-Bucket importieren](#)
- [Schritt 4: Laden Sie Objekte manuell in Ihren Verzeichnis-Bucket hoch](#)

- [Schritt 5: Leeren Sie Ihren Verzeichnis-Bucket](#)
- [Schritt 6: Löschen Sie Ihren Verzeichnis-Bucket](#)
- [Nächste Schritte](#)

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie eine AWS-Konto bei dem Sie sich anmelden können als AWS Identity and Access Management (IAM) Benutzer mit den richtigen Berechtigungen.

Teilschritte

- [Erstellen Sie ein AWS-Konto](#)
- [Erstellen Sie einen IAM Benutzer in Ihrem AWS-Konto \(Konsole\)](#)
- [Erstellen Sie eine IAM Richtlinie und fügen Sie sie einem IAM Benutzer oder einer Rolle \(Konsole\) hinzu](#)

Erstellen Sie ein AWS-Konto


Um dieses Tutorial abzuschließen, benötigen Sie ein AWS-Konto. Wenn du dich anmeldest für AWS, dein AWS-Konto ist automatisch für alle Dienste angemeldet in AWS, einschließlich Amazon S3. Berechnet werden Ihnen aber nur die Services, die Sie nutzen. Weitere Informationen zur Preisgestaltung finden Sie unter [S3-Preise](#).

Erstellen Sie einen IAM Benutzer in Ihrem AWS-Konto (Konsole)

AWS Identity and Access Management (IAM) ist ein AWS-Service das hilft Administratoren, den Zugriff auf sicher zu kontrollieren AWS Ressourcen schützen. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um auf Objekte zuzugreifen und Verzeichnis-Buckets in S3 Express One Zone zu verwenden. Sie können es ohne IAM zusätzliche Kosten nutzen.

Standardmäßig sind Benutzer nicht berechtigt, auf Verzeichnis-Buckets zuzugreifen und S3 Express One Zone-Operationen auszuführen. Um Zugriffsberechtigungen für Verzeichnis-Buckets und S3 Express One Zone-Operationen IAM zu gewähren, können Sie Benutzer oder Rollen erstellen und diesen Identitäten Berechtigungen zuweisen. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie unter [IAMBenutzer erstellen \(Konsole\)](#) im IAMBenutzerhandbuch. Weitere Informationen zum Erstellen einer IAM Rolle finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM Benutzer](#) im IAMBenutzerhandbuch.

Der Einfachheit halber wird in diesem Tutorial ein IAM Benutzer erstellt und verwendet. Denken Sie nach Abschluss dieses Tutorials an [Löschen Sie den IAM Benutzer](#). Für den produktiven Einsatz empfehlen wir, dass Sie die [bewährten Sicherheitsmethoden IAM im IAM](#) Benutzerhandbuch befolgen. Eine bewährte Methode sieht vor, dass menschliche Benutzer für den Zugriff einen Verbund mit einem Identitätsanbieter verwenden AWS mit temporären Anmeldeinformationen. Eine weitere bewährte Methode besteht darin, Workloads die Verwendung temporärer Anmeldeinformationen mit IAM Rollen für den Zugriff vorzuschreiben AWS. Um mehr über die Verwendung von zu erfahren AWS IAM Identity Center Informationen zum Erstellen von Benutzern mit temporären Anmeldeinformationen finden Sie unter [Erste Schritte](#) im AWS IAM Identity Center Benutzerleitfaden.

 Warning

IAM Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

Erstellen Sie eine IAM Richtlinie und fügen Sie sie einem IAM Benutzer oder einer Rolle (Konsole) hinzu

Standardmäßig haben Benutzer keine Berechtigungen für Verzeichnis-Buckets und S3-Express-One-Zone-Vorgänge. Um Zugriffsberechtigungen für Verzeichnis-Buckets IAM zu gewähren, können Sie Benutzer, Gruppen oder Rollen erstellen und diesen Identitäten Berechtigungen zuweisen. Verzeichnis-Buckets sind die einzige Ressource, die Sie in Bucket-Richtlinien oder IAM Identitätsrichtlinien für den Zugriff auf S3 Express One Zone aufnehmen können.

Um regionale API Endpunktoperationen (Operationen auf Bucket-Ebene oder auf Kontrollebene) mit S3 Express One Zone zu verwenden, verwenden Sie das IAM Autorisierungsmodell, das kein Sitzungsmanagement beinhaltet. Berechtigungen werden für Aktionen einzeln erteilt. Um zonale API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) [CreateSession](#) zu verwenden, erstellen und verwalten Sie Sitzungen, die für die Autorisierung von Datenanforderungen mit geringer Latenz optimiert sind. Um ein Sitzungstoken abzurufen und zu verwenden, müssen Sie die `s3express:CreateSession`-Aktion für Ihren Verzeichnis-Bucket in einer identitätsbasierten Richtlinie oder einer Bucket-Richtlinie zulassen. Wenn Sie in der Amazon S3-Konsole auf S3 Express One Zone zugreifen, verwenden Sie AWS Befehlszeilenschnittstelle (AWS CLI), oder mit dem AWS SDKs, S3 Express One Zone erstellt in Ihrem Namen eine Sitzung. Weitere Informationen finden

Sie unter [CreateSessionAutorisierung](#) und [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).


Um eine IAM Richtlinie zu erstellen und die Richtlinie einem IAM Benutzer (oder einer Rolle) zuzuordnen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Management Console.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie JSON.
5. Kopieren Sie die nachstehende Richtlinie in das Fenster des Richtlinien-Editors. Bevor Sie Verzeichnis-Buckets erstellen oder S3 Express One Zone verwenden können, müssen Sie Ihrem AWS Identity and Access Management (IAM) Rolle oder Benutzer. Diese Beispielrichtlinie ermöglicht den Zugriff auf den CreateSession API Vorgang (zur Verwendung mit anderen Operationen auf Zonen- oder Objektebene) und alle API Operationen auf regionaler Endpunktebene (Bucket-Ebene). API Diese Richtlinie ermöglicht die Verwendung des CreateSession API Vorgangs mit allen Verzeichnis-Buckets, aber die regionalen API Endpunktoperationen sind nur für die Verwendung mit dem angegebenen Verzeichnis-Bucket zulässig. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessRegionalEndpointAPIs",
      "Effect": "Allow",
      "Action": [
        "s3express:DeleteBucket",
        "s3express:DeleteBucketPolicy",
        "s3express:CreateBucket",
        "s3express:PutBucketPolicy",
        "s3express:GetBucketPolicy",
        "s3express:ListAllMyDirectoryBuckets"
      ],
      "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-name--azid--x-s3/*"
    }
  ]
}
```

```
    },  
    {  
      "Sid": "AllowCreateSession",  
      "Effect": "Allow",  
      "Action": "s3express:CreateSession",  
      "Resource": "*"   
    }  
  ]  
}
```

6. Wählen Sie Weiter.
7. Geben Sie der Richtlinie einen Namen.

 Note

Bucket-Tags werden für S3 Express One Zone nicht unterstützt.

8. Wählen Sie Create Policy.
9. Nachdem Sie eine IAM Richtlinie erstellt haben, können Sie sie einem IAM Benutzer zuordnen. Wählen Sie im Navigationsbereich Richtlinien.
10. Geben Sie in der Suchleiste den Namen Ihrer Richtlinie ein.
11. Wählen Sie im Menü Aktionen die Option Anhängen aus.
12. Wählen Sie unter Nach Entitätstyp filtern IAM Benutzer oder Rollen aus.
13. Geben Sie im Suchfeld den Namen des Benutzers oder der Rolle ein, den Sie verwenden möchten.
14. Wählen Sie Richtlinie anfügen aus.

Schritt 1: Konfigurieren Sie einen VPC Gateway-Endpunkt

Sie können über Gateway-Endpunkte der virtuellen privaten Cloud (VPC) sowohl auf zonale als auch auf regionale API Operationen zugreifen. Gateway-Endpunkte können es ermöglichen, dass der Verkehr die S3 Express One Zone erreicht, ohne ein Gateway zu passieren. NAT Wir empfehlen dringend, Gateway-Endpunkte zu verwenden, da sie bei der Arbeit mit S3 Express One Zone den optimalsten Netzwerkpfad bieten. Sie können von Ihrem Computer aus auf S3 Express One Zone-Verzeichnis-Buckets zugreifen, VPC ohne ein Internet-Gateway oder ein NAT Gerät für Sie zu haben VPC, und das ohne zusätzliche Kosten. Gehen Sie wie folgt vor, um einen Gateway-Endpunkt

zu konfigurieren, der eine Verbindung zu S3 Express One Zone-Speicherklassenobjekten und Verzeichnis-Buckets herstellt.

Für den Zugriff auf S3 Express One Zone verwenden Sie regionale und zonale Endpunkte, die sich von den standardmäßigen Amazon-S3-Endpunkten unterscheiden. Je nachdem, welchen Amazon S3 API S3-Betrieb Sie verwenden, ist entweder ein zonaler oder ein regionaler Endpunkt erforderlich. Eine vollständige Liste der unterstützten API Operationen nach Endpunkttyp finden Sie unter [Von S3 Express One Zone unterstützte API Operationen](#). Sie müssen über einen Gateway-Endpunkt der virtuellen privaten Cloud (VPC) auf sowohl zonale als auch regionale Endpunkte zugreifen.

Gehen Sie wie folgt vor, um einen Gateway-Endpunkt zu erstellen, der eine Verbindung zu S3 Express One Zone-Speicherklassenobjekten und Verzeichnis-Buckets herstellt.

Um einen VPC Gateway-Endpunkt zu konfigurieren

1. Öffnen Sie die VPC Amazon-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im seitlichen Navigationsbereich unter Virtual Private Cloud die Option Endpoints aus.
3. Wählen Sie Endpunkt erstellen aus.
4. Erstellen Sie einen Namen für Ihren Endpunkt.
5. Wählen Sie als Servicekategorie AWS-Services.
6. Suchen Sie unter Dienste mit dem Filter Type=Gateway und wählen Sie dann die Optionsschaltfläche neben `com.amazonaws.region.s3express`.
7. Wählen Sie für die VPC, VPC in der der Endpunkt erstellt werden soll.
8. Wählen Sie für Route tables (Routing-Tabellen) die Routing-Tabellen, die von dem Endpunkt verwendet werden sollen. Amazon fügt VPC automatisch eine Route hinzu, die den für den Service bestimmten Verkehr an die Netzwerkschnittstelle des Endpunkts weiterleitet.
9. Wählen Sie für Policy die Option Vollzugriff aus, um alle Operationen aller Principals auf allen Ressourcen über den VPC Endpunkt zuzulassen. Wählen Sie andernfalls Benutzerdefiniert, um eine VPC Endpunktrichtlinie anzuhängen, die die Berechtigungen steuert, die Prinzipale haben, um Aktionen an Ressourcen über den VPC Endpunkt auszuführen.
10. Wählen Sie Endpunkt erstellen aus.

Nachdem Sie einen Gateway-Endpunkt erstellt haben, können Sie regionale API Endpunkte und zonale API Endpunkte verwenden, um auf Amazon S3 Express One Zone-Speicherklassenobjekte und Verzeichnis-Buckets zuzugreifen.

Schritt 2: Erstellen Sie einen Verzeichnis-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

Note

Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Für eine Liste von Amazon S3 AWS-Regionen, siehe [AWS-Service Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

Anschließend wird die Seite Bucket erstellen geöffnet.


5. Sehen Sie sich unter Allgemeine Konfiguration die AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Verzeichnis aus.

Note

- Wenn Sie eine Region ausgewählt haben, die keine Verzeichnis-Buckets unterstützt, verschwindet die Option Bucket-Typ und der Bucket-Typ ist standardmäßig ein Allzweck-Bucket. Um einen Verzeichnis-Bucket zu erstellen, müssen Sie eine unterstützte Region auswählen. Eine Liste der Regionen, die Directory-Buckets und die Amazon S3 Express One Zone-Speicherklasse unterstützen, finden Sie unter [the section called "Availability Zones und Regionen bei S3 Express One Zone"](#).
- Nach dem Erstellen des Buckets kann der Bucket-Typ nicht mehr geändert werden.


Wählen Sie für Availability Zone eine Availability Zone aus, die für Ihre Computing-Services lokal ist. Eine Liste der Availability Zones, die Directory-Buckets und die Speicherklasse S3 Express

One Zone unterstützen, finden Sie unter [the section called “Availability Zones und Regionen bei S3 Express One Zone”](#)

 Note

Die Availability Zone kann nicht geändert werden, nachdem der Bucket erstellt wurde.

7. Aktivieren Sie unter Availability Zone das Kontrollkästchen, um zu bestätigen, dass Ihre Daten im Falle eines Ausfalls der Availability Zone möglicherweise nicht verfügbar sind oder verloren gehen.

 Important

Obwohl Verzeichnis-Buckets auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert werden, speichern Verzeichnis-Buckets Daten nicht redundant über Availability Zones hinweg.

8. Geben Sie unter Bucket-Name einen Namen für Ihren Verzeichnis-Bucket ein.

Die folgenden Benennungsregeln gelten für Verzeichnis-Buckets.


- Seien Sie innerhalb der Auserwählten einzigartig AWS-Region und Availability Zone.
- Der Name muss zwischen 3 (min) und 63 (maximal) Zeichen lang sein, einschließlich des Suffixes.
- Nur aus Kleinbuchstaben, Zahlen und Bindestrichen bestehen.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.
- Muss das folgende Suffix enthalten: `--azid--x-s3`
- Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
- Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#).
- Bucket-Namen dürfen nicht mit dem Suffix `--o1-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [So](#)

[verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets.](#)

- Bucket-Namen dürfen nicht mit dem Suffix `.mr.ap` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter [Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten](#).

Dem Basisnamen, den Sie angeben, wenn Sie mit der Konsole einen Verzeichnis-Bucket erstellen, wird automatisch ein Suffix hinzugefügt. Dieses Suffix enthält die ID der Availability Zone, die Sie ausgewählt haben.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

 **Important**

Nehmen Sie keine vertraulichen Informationen wie Kontonummern in den Bucket-Namen auf. Der Bucket-Name ist in dem Punkt `sightbarURLs`, der auf die Objekte im Bucket verweist.

9. Unter Object Ownership ist die Einstellung Bucket Owner enforced automatisch aktiviert und alle Zugriffskontrolllisten (ACLs) sind deaktiviert. Kann für Verzeichnis-Buckets nicht ACLs aktiviert werden.

ACLsdeaktiviert

- Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

10. Unter „Einstellungen für öffentlichen Zugriff blockieren“ für diesen Bucket werden alle Einstellungen für „Öffentlichen Zugriff blockieren“ für Ihren Verzeichnis-Bucket automatisch aktiviert. Diese Einstellungen können für Directory-Buckets nicht geändert werden. Weitere

Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

11. Unter Serverseitige Verschlüsselungseinstellungen verwenden Verzeichnis-Buckets serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), um Daten zu verschlüsseln.
12. Wählen Sie Bucket erstellen aus.

Nachdem Sie den Bucket erstellt haben, können Sie ihm Dateien und Ordner hinzufügen. Weitere Informationen finden Sie unter [the section called “Arbeiten mit Objekten in einem Verzeichnis-Bucket”](#).

Der folgende Schritt zeigt, wie Sie die Import-Aktion in der Amazon S3 S3-Konsole verwenden, um Ihren Verzeichnis-Bucket mit Daten zu füllen.

Schritt 3: Daten in einen Verzeichnis-Bucket importieren


Um diesen Schritt abschließen zu können, benötigen Sie einen Allzweck-Bucket, der Objekte enthält und sich in demselben befindet AWS-Region als Ihren Verzeichnis-Bucket.

Nachdem Sie einen Verzeichnis-Bucket in Amazon S3 erstellt haben, können Sie den neuen Bucket mit Daten füllen, indem Sie die Import-Aktion in der Amazon S3 S3-Konsole verwenden. Import vereinfacht das Kopieren von Daten in Verzeichnis-Buckets, indem Sie ein Präfix oder einen Allzweck-Bucket auswählen können, aus dem Daten importiert werden sollen, ohne alle zu kopierenden Objekte einzeln angeben zu müssen. Beim Import werden S3 Batch Operations verwendet, bei denen die Objekte im ausgewählten Präfix oder im Allzweck-Bucket kopiert werden. Sie können den Fortschritt des Importkopierauftrags auf der Seite mit den Auftragsdetails für S3 Batch Operations überwachen.

Um die Importaktion zu verwenden

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
3. Wählen Sie im linken Navigationsbereich Buckets und dann die Registerkarte Verzeichnis-Buckets aus. Wählen Sie den Verzeichnis-Bucket aus, in den Sie Objekte importieren möchten.

4. Wählen Sie Importieren aus.
5. Geben Sie unter Quelle den Allzweck-Bucket (oder den Bucket-Pfad einschließlich Präfix) ein, der die Objekte enthält, die Sie importieren möchten. Um einen vorhandenen Allzweck-Bucket aus einer Liste auszuwählen, wählen Sie S3 durchsuchen.
6. Im Bereich Berechtigungen können Sie festlegen, dass eine IAM Rolle automatisch generiert wird. Alternativ können Sie eine IAM Rolle aus einer Liste auswählen oder direkt eine IAM Rolle ARN eingeben.
 - Damit Amazon S3 in Ihrem Namen eine neue IAM Rolle erstellen kann, wählen Sie Neue IAM Rolle erstellen.

 Note

Wenn Ihre Quellobjekte serverseitig verschlüsselt sind mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), wählen Sie nicht die Option Neue IAM Rolle erstellen. Geben Sie stattdessen eine bestehende IAM Rolle an, die über die `kms:Decrypt` entsprechende Berechtigung verfügt.

Amazon S3 verwendet diese Berechtigung, um Ihre Objekte zu entschlüsseln.

Während des Importvorgangs verschlüsselt Amazon S3 diese Objekte dann erneut, indem es serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwendet.

- Um eine bestehende IAM Rolle aus einer Liste auszuwählen, wählen Sie Aus vorhandenen Rollen auswählen. IAM
 - Um eine bestehende IAM Rolle durch Eingabe ihres Amazon-Ressourcennamens (ARN) anzugeben, wählen Sie IAM Rolle eingeben ARN und geben Sie dann das ARN in das entsprechende Feld ein.
7. Überprüfen Sie die Informationen, die in den Abschnitten Ziel und Kopierte Objekteinstellungen angezeigt werden. Wenn die Informationen im Bereich Ziel korrekt sind, wählen Sie Import, um den Kopierauftrag zu starten.

Die Amazon-S3-Konsole zeigt den Status Ihres neuen Auftrags auf der Seite Batch-Operationen an. Für weitere Informationen über den Auftrag klicken Sie auf das Optionsfeld neben dem Auftragsnamen und wählen dann im Menü Aktionen die Option Details anzeigen. Um den Verzeichnis-Bucket zu öffnen, in den die Objekte importiert werden, wählen Sie Importziel anzeigen.

Schritt 4: Laden Sie Objekte manuell in Ihren Verzeichnis-Bucket hoch

Sie können Objekte auch manuell in Ihren Verzeichnis-Bucket hochladen.

Um Objekte manuell hochzuladen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste in der oberen rechten Ecke der Seite den Namen der aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
3. Wählen Sie im linken Navigationsbereich Buckets aus.
4. Wählen Sie die Registerkarte Verzeichnis-Buckets.
5. Wählen Sie den Namen des Buckets aus, in den Sie Ihre Ordner oder Dateien hochladen möchten.

Note

Wenn Sie denselben Verzeichnis-Bucket ausgewählt haben, den Sie in den vorherigen Schritten dieses Tutorials verwendet haben, enthält Ihr Verzeichnis-Bucket die Objekte, die mit dem Import-Tool hochgeladen wurden. Beachten Sie, dass diese Objekte jetzt in der S3 Express One Zone-Speicherklasse gespeichert sind.

6. Wählen Sie in der Objektliste die Option Hochladen aus.
7. Führen Sie auf der Upload-Seite einen der folgenden Schritte aus:
 - Ziehen Sie Dateien und Ordner per Drag & Drop in den gepunkteten Upload-Bereich.
 - Wählen Sie Dateien hinzufügen oder Ordner hinzufügen, wählen Sie die hochzuladenden Dateien oder Ordner aus und wählen Sie dann Öffnen oder Hochladen.
8. Wählen Sie unter Prüfsummen die Prüfsummenfunktion aus, die Sie verwenden möchten.

Note

Wir empfehlen die Verwendung CRC32 und CRC32C für die beste Leistung mit der Speicherklasse S3 Express One Zone. Weitere Informationen finden Sie unter [Bewährte Methoden für zusätzliche S3-Prüfsummen](#).

(Optional) Wenn Sie ein einzelnes Objekt hochladen, das weniger als 16 MB groß ist, können Sie auch einen vorab berechneten Prüfsummenwert angeben. Wenn Sie einen vorberechneten Wert angeben, vergleicht Amazon S3 ihn mit dem Wert, den es mithilfe der ausgewählten Prüfsummenfunktion berechnet. Wenn die Werte nicht übereinstimmen, startet der Upload nicht.

- Die Optionen in den Abschnitten „Berechtigungen“ und „Eigenschaften“ werden automatisch auf die Standardeinstellungen gesetzt und können nicht geändert werden. Block Public Access ist automatisch aktiviert, und S3 Versioning und S3 Object Lock können für Directory-Buckets nicht aktiviert werden.

(Optional) Wenn Sie Ihren Objekten Metadaten in Schlüssel-Wert-Paaren hinzufügen möchten, erweitern Sie den Abschnitt Eigenschaften und wählen Sie dann im Abschnitt Metadaten die Option Metadaten hinzufügen aus.

- Um die aufgelisteten Dateien und Ordner hochzuladen, wählen Sie Hochladen.

Amazon S3 lädt Ihre Objekte und Ordner hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: Status eine Erfolgsmeldung angezeigt.

Sie haben erfolgreich einen Verzeichnis-Bucket erstellt und Objekte in Ihren Bucket hochgeladen.

Schritt 5: Leeren Sie Ihren Verzeichnis-Bucket

Sie können Ihren Amazon S3 S3-Verzeichnis-Bucket mithilfe der Amazon S3 S3-Konsole leeren.

Um einen Verzeichnis-Bucket zu leeren

- Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
- Wählen Sie in der Navigationsleiste in der oberen rechten Ecke der Seite den Namen der aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
- Wählen Sie im linken Navigationsbereich Buckets aus.
- Wählen Sie die Registerkarte Verzeichnis-Buckets.
- Wählen Sie das Optionsfeld neben dem Namen des Buckets aus, den Sie leeren möchten, und wählen Sie dann Leer aus.

6. Bestätigen Sie auf der Seite Empty bucket (Bucket leeren), dass Sie den Bucket leeren möchten, indem Sie **permanently delete** in das Textfeld eingeben und dann Empty (Leeren) auswählen.
7. Überwachen Sie den Fortschritt beim Entleeren des Buckets auf der Seite Leerer Bucket: Status.

Schritt 6: Löschen Sie Ihren Verzeichnis-Bucket

Nachdem Sie Ihren Verzeichnis-Bucket geleert und alle laufenden mehrteiligen Uploads abgebrochen haben, können Sie Ihren Bucket mithilfe der Amazon S3 S3-Konsole löschen.

Um einen Verzeichnis-Bucket zu löschen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste in der oberen rechten Ecke der Seite den Namen der aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
3. Wählen Sie im linken Navigationsbereich Buckets aus.
4. Wählen Sie die Registerkarte Verzeichnis-Buckets.
5. Wählen Sie in der Liste der Directory-Buckets das Optionsfeld neben dem Bucket aus, den Sie löschen möchten.
6. Wählen Sie Löschen.
7. Geben Sie auf der Seite „Bucket löschen“ den Namen des Buckets in das Textfeld ein, um das Löschen Ihres Buckets zu bestätigen.

Important

Das Löschen eines Verzeichnis-Buckets kann nicht rückgängig gemacht werden.

8. Um Ihren Verzeichnis-Bucket zu löschen, wählen Sie Bucket löschen aus.

Nächste Schritte

In diesem Tutorial haben Sie gelernt, wie Sie einen Verzeichnis-Bucket erstellen und die S3 Express One Zone-Speicherklasse verwenden. Nach Abschluss dieses Tutorials können Sie sich mit

verwandten Themen befassen AWS Dienste zur Verwendung mit der Speicherklasse S3 Express One Zone.

Sie können Folgendes verwenden AWS-Services mit der Speicherklasse S3 Express One Zone zur Unterstützung Ihres spezifischen Anwendungsfalls mit niedriger Latenz.

- [Amazon Elastic Compute Cloud \(AmazonEC2\)](#) — Amazon EC2 bietet sichere und skalierbare Rechenkapazität in der AWS Cloud. Durch die Verwendung von Amazon EC2 müssen Sie nicht im Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwenden EC2, um so viele oder so wenige virtuelle Server wie nötig zu starten, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten.
- [AWS Lambda](#) – Lambda ist ein Computing-Service, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Sie konfigurieren Benachrichtigungseinstellungen für einen Bucket und erteilen die Amazon-S3-Berechtigung zum Aufrufen einer Funktion in der ressourcenbasierten Berechtigungsrichtlinie der Funktion.
- [Amazon Elastic Kubernetes Service \(AmazonEKS\)](#) — Amazon EKS ist ein verwalteter Service, der die Installation, den Betrieb und die Wartung Ihres eigenen Services überflüssig macht. Kubernetes Kontrollebene ein AWS. [Kubernetes](#) ist ein Open-Source-System, das die Verwaltung, Skalierung und Bereitstellung containerisierter Anwendungen automatisiert.
- [Amazon Elastic Container Service \(AmazonECS\)](#) — Amazon ECS ist ein vollständig verwalteter Container-Orchestrierungsservice, mit dem Sie containerisierte Anwendungen einfach bereitstellen, verwalten und skalieren können.
- [Amazon EMR — Amazon](#) EMR ist eine verwaltete Cluster-Plattform, die den Betrieb von Big-Data-Frameworks vereinfacht, wie z. B. Apache Hadoop und Apache Spark on AWS um riesige Datenmengen zu verarbeiten und zu analysieren.
- [Amazon Athena](#) — Athena ist ein interaktiver Abfrageservice, der es einfach macht, Daten mithilfe von Standards direkt in Amazon S3 zu analysieren. [SQL](#) Sie können Athena auch verwenden, um Datenanalysen interaktiv auszuführen, indem Sie Apache Spark ohne Ressourcen planen, konfigurieren oder verwalten zu müssen. Wenn du rennst Apache Spark Bewerbungen auf Athena reichen Sie ein Spark Code zur Verarbeitung und direkt die Ergebnisse erhalten.
- [AWS Glue-Datenkatalog](#) — AWS Glue ist ein serverloser Datenintegrationsdienst, der es Analytics-Benutzern leicht macht, Daten aus mehreren Quellen zu entdecken, aufzubereiten, zu verschieben und zu integrieren. Sie können Folgendes verwenden ... AWS Glue für Analytik, maschinelles Lernen und Anwendungsentwicklung. AWS Glue Data Catalog ist ein zentrales Repository, das Metadaten zu den Datensätzen Ihres Unternehmens speichert. Es dient als Index für den Standort, das Schema und die Laufzeitmetriken Ihrer Datenquellen.

- [Amazon SageMaker Runtime Model Training](#) — Amazon SageMaker Runtime ist ein vollständig verwalteter Service für maschinelles Lernen. Mit SageMaker Runtime können Datenwissenschaftler und Entwickler schnell und einfach Modelle für maschinelles Lernen erstellen und trainieren und sie dann direkt in einer produktionsbereiten gehosteten Umgebung bereitstellen.

Weitere Informationen zu S3 Express One Zone finden Sie unter [Was ist S3 Express One Zone?](#) und [wie unterscheidet sich S3 Express One Zone?](#) .

Networking für S3 Express One Zone

Für den Zugriff auf Amazon-S3-Express-One-Zone-Objekte und Verzeichnis-Buckets verwenden Sie zonale Endpunkte, die sich von den standardmäßigen Amazon-S3-Endpunkten unterscheiden. Je nachdem, welchen S3-API-Vorgang Sie verwenden, ist entweder ein zonaler oder ein regionaler Endpunkt erforderlich. Eine vollständige Liste der API-Operationen nach Endpunkttyp finden Sie unter [API von S3 Express One Zone unterstützte Operationen](#).

Sie können über Gateway Virtual-Private-Cloud (VPC)-Endpunkte auf zonale und regionale Endpunkte zugreifen. Informationen zur Konfiguration von Gateway-VPC-Endpunkten finden Sie unter [the section called “Konfiguration von VPC-Gateway-Endpunkten”](#).

In den folgenden Themen werden die Netzwerkanforderungen für den Zugriff auf S3 Express One Zone über einen Gateway-VPC-Endpunkt beschrieben.

Themen

- [Endpunkte](#)
- [Konfiguration von VPC-Gateway-Endpunkten](#)

Endpunkte

Sie können über Gateway-VPC-Endpunkte von Ihrer VPC aus auf Objekte der Speicherklasse Amazon S3 Express One und Verzeichnis-Buckets zugreifen. S3 Express One Zone verwendet regionale und zonale API-Endpunkte. Je nachdem, welche Amazon-S3-API-Operation Sie verwenden, ist entweder ein regionaler oder ein zonaler Endpunkt erforderlich. Für die Nutzung von Gateway-Endpunkten fallen keine zusätzlichen Gebühren an.

API-Vorgänge auf Bucket-Ebene (Steuerebene) sind über einen regionalen Endpunkt verfügbar und werden als API-Vorgänge für regionale Endpunkte bezeichnet. Beispiele für API-Operationen

für regionalen Endpunkte sind `CreateBucket` und `DeleteBucket`. Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie eine einzelne Availability Zone aus, in der Ihr Verzeichnis-Bucket erstellt wird. Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie mithilfe von API-Operationen für zonale Endpunkte die Objekte in Ihrem Verzeichnis-Bucket hochladen und verwalten.

API-Operationen auf Objektebene (oder Datenebene) sind über zonale Endpunkte verfügbar und werden als API-Operationen für zonale Endpunkte bezeichnet. Beispiele für API-Operationen für zonale Endpunkte sind `CreateSession` und `PutObject`.

Die folgende Tabelle zeigt die regionalen und zonalen API-Endpunkte, die für jede Region und Availability Zone verfügbar sind.

Konfiguration von VPC-Gateway-Endpunkten

Gehen Sie wie folgt vor, um einen Gateway-Endpunkt zu erstellen, der eine Verbindung zu Objekten der Speicherklasse Amazon S3 Express One Zone und Verzeichnis-Buckets herstellt.

So konfigurieren Sie einen Gateway-VPC-Endpunkt

1. Öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie Endpunkt erstellen.
4. Erstellen Sie einen Namen für Ihren Endpunkt.
5. Wählen Sie für Servicekategorie die Option AWS-Services aus.
6. Fügen Sie für Services den Filter Typ=Gateway hinzu und wählen Sie dann das Optionsfeld neben `com.amazonawsregion.s3express`.
7. Wählen Sie für VPC die VPC, in der der Endpunkt erstellt werden soll.
8. Wählen Sie für Route tables (Routing-Tabellen) die Routing-Tabellen, die von dem Endpunkt verwendet werden sollen. Amazon VPC fügt automatisch eine Route hinzu, die den für den Service bestimmten Datenverkehr auf die Netzwerkschnittstelle des Endpunkts verweist.
9. Wählen Sie für Richtlinie Vollzugriff, um alle Operationen aller Prinzipale auf allen Ressourcen über den VPC-Endpunkt zuzulassen. Wählen Sie andernfalls Benutzerdefiniert, um eine VPC-Endpunktrichtlinie anzufügen, die die Berechtigungen steuert, die Prinzipale zum Ausführen von Aktionen für Ressourcen über den VPC-Endpunkt haben.
10. (Optional) Sie fügen ein Tag hinzu, indem Sie Neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.

11. Wählen Sie Endpunkt erstellen aus.

Nachdem Sie einen Gateway-Endpunkt erstellt haben, können Sie regionale API-Endpunkte und zonale API-Endpunkte verwenden, um auf Objekte in der Speicherklasse Amazon S3 Express One Zone und Verzeichnis-Buckets zuzugreifen.

Verzeichnis-Buckets

Es gibt zwei Arten von Amazon-S3-Buckets: Allzweck-Buckets und Verzeichnis-Buckets. Wählen Sie den Bucket-Typ, der am besten zu Ihren Anwendungs- und Leistungsanforderungen passt:

- Allzweck-Buckets sind der ursprüngliche S3-Bucket-Typ und werden für die meisten Anwendungsfälle und Zugriffsmuster empfohlen. Allzweck-Buckets ermöglichen auch Objekte, die in allen Speicherklassen gespeichert werden, mit Ausnahme von S3 Express One Zone.
- Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone. Dies wird empfohlen, wenn Ihre Anwendung leistungsempfindlich ist und von PUT- und GET-Latenzen im einstelligen Millisekundenbereich profitiert.

Verzeichnis-Buckets werden für Workloads oder leistungskritische Anwendungen verwendet, die eine konsistente Latenzzeit im einstelligen Millisekundenbereich erfordern. Verzeichnis-Buckets organisieren Daten hierarchisch in Verzeichnissen, im Gegensatz zu der flachen Speicherstruktur von Allzweck-Buckets. Es gibt keine Präfixbeschränkungen für Verzeichnis-Buckets und einzelne Verzeichnisse können horizontal skaliert werden.

Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone, die Daten auf mehreren Geräten innerhalb einer einzigen Availability Zone speichert, Daten jedoch nicht redundant über Availability Zones hinweg speichert. Wenn Sie einen Directory-Bucket erstellen, empfehlen wir Ihnen, eine Availability Zone AWS-Region und eine Availability Zone anzugeben, die lokal für Ihre Amazon EC2-, Amazon Elastic Kubernetes Service- oder Amazon Elastic Container Service (Amazon ECS) Compute-Instances sind, um die Leistung zu optimieren.

Sie können in jedem Ihrer Verzeichnis-Buckets bis zu 10 Verzeichnis-Buckets erstellen AWS-Konten, wobei die Anzahl der Objekte, die Sie in einem Bucket speichern können, unbegrenzt ist. Ihr Bucket-Kontingent wird auf jede Region in Ihrem AWS-Konto angewendet. Wenn Ihre Anwendung eine Erhöhung dieses Limits erfordert, wenden Sie sich an AWS Support. Weitere Informationen finden Sie in der [Service Quotas Quotas-Konsole](#).

Important

Verzeichnis-Buckets, die über einen Zeitraum von mindestens 90 Tagen keine Anforderungsaktivität aufweisen, gehen in einen inaktiven Status über. Im inaktiven Zustand ist ein Verzeichnis-Bucket vorübergehend für Lese- und Schreibvorgänge nicht zugänglich. Inaktive Buckets behalten alle Speicher-, Objekt- und Bucket-Metadaten bei. Für inaktive Buckets fallen bestehende Speichergebühren an. Wenn Sie eine Zugriffsanfrage für einen inaktiven Bucket stellen, wechselt der Bucket in den aktiven Status, normalerweise innerhalb weniger Minuten. Während dieser Übergangsphase geben Lese- und Schreibvorgänge einen 503 (Service Unavailable) HTTP-Fehlercode zurück.

Die folgenden Themen enthalten Informationen zu Verzeichnis-Buckets. Weitere Informationen über Allzweck-Buckets finden Sie unter [Bucket-Übersicht](#).

Themen

- [Availability Zones](#)
- [Namen von Verzeichnis-Buckets](#)
- [Verzeichnisse](#)
- [Schlüsselnamen](#)
- [Zugriffsverwaltung](#)
- [Arbeiten mit Verzeichnis-Buckets](#)
- [Regeln für die Benennung von Verzeichnis-Buckets](#)
- [Erstellen eines Verzeichnis-Buckets](#)
- [Anzeigen von Verzeichnis-Bucket-Eigenschaften](#)
- [Verwalten von Bucket-Richtlinien für Verzeichnis-Buckets](#)
- [Leeren eines Verzeichnis-Buckets](#)
- [Löschen eines Verzeichnis-Buckets](#)
- [Auflisten von Verzeichnis-Buckets](#)
- [Verwendung HeadBucket mit Verzeichnis-Buckets](#)

Availability Zones

Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie die Availability Zone und die AWS-Region.

Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone, die für die Verwendung durch leistungssensitive Anwendungen konzipiert wurde. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet.

Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das [Amazon S3 Service Level Agreement](#) unterstützt. Weitere Informationen finden Sie unter [Einzelne Availability Zone](#).

Namen von Verzeichnis-Buckets

Der Name eines Verzeichnis-Buckets besteht aus einem Basisnamen, den Sie angeben, und einem Suffix, das die ID der Availability Zone enthält, in der sich Ihr Bucket befindet. Die Namen von Verzeichnis-Buckets müssen das folgende Format haben und den Benennungsregeln für Verzeichnis-Buckets entsprechen:

```
bucket-base-name--azid--x-s3
```

Der folgende Verzeichnis-Bucket-Name enthält beispielsweise die Availability Zone-ID usw2-az1:

```
bucket-base-name--usw2-az1--x-s3
```

Weitere Informationen finden Sie unter [Regeln für die Benennung von Verzeichnis-Buckets](#).

Verzeichnisse

Verzeichnis-Buckets organisieren Daten hierarchisch in Verzeichnissen, im Gegensatz zu der flachen Sortierstruktur von Allzweck-Buckets. Jeder S3-Verzeichnis-Bucket kann Hunderttausende von Transaktionen pro Sekunde (TPS) unterstützen, unabhängig von der Anzahl der Verzeichnisse innerhalb des Buckets.

Bei einem hierarchischen Namespace ist das Trennzeichen im Objektschlüssel wichtig. Das einzige unterstützte Trennzeichen ist der Schrägstrich (/). Verzeichnisse werden durch Trennzeichengrenzen bestimmt. Beispielsweise führt der Objektschlüssel `dir1/dir2/file1.txt` dazu, dass die Verzeichnisse `dir1/` und `dir2/` automatisch erstellt werden und das Objekt `file1.txt` dem `/dir2/`-Verzeichnis im Pfad `dir1/dir2/file1.txt` hinzugefügt wird.

Das Indizierungsmodell für Verzeichnis-Buckets gibt unsortierte Ergebnisse für den ListObjectsV2-API-Vorgang zurück. Wenn Sie Ihre Ergebnisse auf einen Unterabschnitt Ihres Buckets beschränken müssen, können Sie im `prefix`-Parameter einen Unterverzeichnispfad angeben, z. B. `prefix=dir1/`.

Schlüsselnamen

Bei Verzeichnis-Buckets werden Unterverzeichnisse, die mehreren Objektschlüsseln gemeinsam sind, mit dem ersten Objektschlüssel erstellt. Zusätzliche Objektschlüssel für dasselbe Unterverzeichnis verwenden das zuvor erstellte Unterverzeichnis. Dieses Modell bietet Ihnen Flexibilität bei der Auswahl von Objektschlüsseln, die für die Anwendung am besten geeignet sind, und unterstützt sowohl dünn besetzte als auch dichte Verzeichnisse.

Zugriffsverwaltung

Bei Verzeichnis-Buckets sind alle Einstellungen für das Blockieren des öffentlichen Zugriffs auf Bucket-Ebene standardmäßig aktiviert. Die S3-Objekt-Eigentümerschaft ist auf „Vom Bucket-Eigentümer erzwungen“ festgelegt und alle Zugriffssteuerungslisten sind deaktiviert. Diese Einstellungen können nicht geändert werden.

Standardmäßig haben Benutzer keine Berechtigungen für Verzeichnis-Buckets und S3-Express-One-Zone-Vorgänge. Um Zugriffsberechtigungen für Verzeichnis-Buckets zu gewähren, können Sie IAM verwenden, um Benutzer, Gruppen oder Rollen zu erstellen und diesen Identitäten Berechtigungen zuzuweisen. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

Arbeiten mit Verzeichnis-Buckets

Weitere Informationen zum Arbeiten mit Verzeichnis-Buckets finden Sie in den folgenden Themen.

Themen

- [Regeln für die Benennung von Verzeichnis-Buckets](#)
- [Erstellen eines Verzeichnis-Buckets](#)
- [Anzeigen von Verzeichnis-Bucket-Eigenschaften](#)
- [Verwalten von Bucket-Richtlinien für Verzeichnis-Buckets](#)
- [Leeren eines Verzeichnis-Buckets](#)
- [Löschen eines Verzeichnis-Buckets](#)

- [Auflisten von Verzeichnis-Buckets](#)
- [Verwendung HeadBucket mit Verzeichnis-Buckets](#)

Regeln für die Benennung von Verzeichnis-Buckets

Wenn Sie einen Verzeichnis-Bucket in Amazon S3 erstellen, gelten dafür die folgenden Benennungsregeln. Benennungsregeln für Allzweck-Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

Ein Verzeichnis-Bucket-Name besteht aus einem Basisnamen, den Sie angeben, und einem Suffix, das die ID der AWS Availability Zone enthält, in der sich Ihr Bucket befindet, und `--x-s3`.

```
base-name--azid--x-s3
```

Der folgende Verzeichnis-Bucket-Name enthält beispielsweise die Availability Zone-ID `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Note

Wenn Sie mithilfe der Konsole einen Verzeichnis-Bucket erstellen, wird dem von Ihnen angegebenen Basisnamen automatisch ein Suffix hinzugefügt. Dieses Suffix enthält die ID der Availability Zone, die Sie ausgewählt haben.

Wenn Sie einen Verzeichnis-Bucket mithilfe von erstellen, müssen API Sie das vollständige Suffix, einschließlich der Availability Zone ID, in Ihrer Anfrage angeben. Eine Liste der Availability Zone finden Sie IDs unter [Availability Zones und Regionen bei S3 Express One Zone](#).

Die folgenden Benennungsregeln gelten für Verzeichnis-Buckets.

- Seien Sie innerhalb der ausgewählten Zone AWS-Region und der Availability Zone einzigartig.
- Der Name muss zwischen 3 (min) und 63 (maximal) Zeichen lang sein, einschließlich des Suffixes.
- Nur aus Kleinbuchstaben, Zahlen und Bindestrichen bestehen.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.
- Muss das folgende Suffix enthalten: `--azid--x-s3`

- Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
- Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#).
- Bucket-Namen dürfen nicht mit dem Suffix `--o1-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets](#).
- Bucket-Namen dürfen nicht mit dem Suffix `.mrp` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter [Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten](#).

Erstellen eines Verzeichnis-Buckets

Um mit der Verwendung der Speicherklasse Amazon S3 Express One Zone zu beginnen, erstellen Sie einen Verzeichnis-Buckets. Die Speicherklasse S3 Express One Zone kann nur mit Verzeichnis-Buckets verwendet werden. Die Speicherklasse S3 Express One Zone unterstützt Anwendungsfälle mit niedriger Latenz und ermöglicht eine schnellere Datenverarbeitung innerhalb einer einzigen Availability Zone. Wenn Ihre Anwendung leistungsempfindlich ist und von PUT- und GET-Latenzen im einstelligen Millisekundenbereich profitiert, empfehlen wir, einen Verzeichnis-Bucket zu erstellen, damit Sie die Speicherklasse S3 Express One Zone verwenden können.

Es gibt zwei Arten von Amazon-S3-Buckets: Allzweck-Buckets und Verzeichnis-Buckets. Sie sollten den Bucket-Typ wählen, der am besten zu Ihren Anwendungs- und Leistungsanforderungen passt. Bei Allzweck-Buckets handelt es sich um den ursprünglichen S3-Bucket-Typ. Allzweck-Buckets werden für die meisten Anwendungsfälle und Zugriffsmuster empfohlen und ermöglichen die Speicherung von Objekten in allen Speicherklassen, mit Ausnahme von S3 Express One Zone. Weitere Informationen über Allzweck-Buckets finden Sie unter [Bucket-Übersicht](#).

Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone, die für Workloads oder leistungskritische Anwendungen konzipiert ist, die eine konsistente Latenz im einstelligen Millisekundenbereich erfordern. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam

mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Wenn Sie einen Verzeichnis-Bucket erstellen, können Sie optional einen angeben AWS-Region und eine Availability Zone, die lokal für Ihre Amazon-EC2, Amazon Elastic Kubernetes Service- oder Amazon Elastic Container Service (AmazonECS) Compute-Instances ist, um die Leistung zu optimieren.

Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das [Amazon S3 Service Level Agreement](#) unterstützt. Weitere Informationen finden Sie unter [Einzelne Availability Zone](#)

Verzeichnis-Buckets organisieren Daten hierarchisch in Verzeichnissen, im Gegensatz zur flachen Speicherstruktur von Allzweck-Buckets. Es gibt keine Präfixbeschränkungen für Verzeichnis-Buckets und einzelne Verzeichnisse können horizontal skaliert werden.

Weitere Information zu Verzeichnis-Buckets finden Sie unter [Verzeichnis-Buckets](#).

Namen von Verzeichnis-Buckets

Die Namen von Verzeichnis-Buckets müssen diesem Format folgen und den Regeln für die Benennung von Verzeichnis-Buckets entsprechen:

```
bucket-base-name--azid--x-s3
```

Der folgende Verzeichnis-Bucket-Name enthält beispielsweise die Availability Zone-ID usw2-az1:

```
bucket-base-name--usw2-az1--x-s3
```

Weitere Informationen zu den Benennungsregeln für Verzeichnis-Buckets finden Sie unter [Regeln für die Benennung von Verzeichnis-Buckets](#).

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

Note

Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Für eine Liste von Amazon S3 AWS-Regionen, siehe [AWS-Service Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

Anschließend wird die Seite Bucket erstellen geöffnet.

5. Sehen Sie sich unter Allgemeine Konfiguration den AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Verzeichnis aus.

Note

- Wenn Sie eine Region ausgewählt haben, die keine Verzeichnis-Buckets unterstützt, verschwindet die Option Bucket-Typ und der Bucket-Typ ist standardmäßig ein Allzweck-Bucket. Um einen Verzeichnis-Bucket zu erstellen, müssen Sie eine unterstützte Region auswählen. Eine Liste der Regionen, die Directory-Buckets und die Amazon S3 Express One Zone-Speicherklasse unterstützen, finden Sie unter [the section called “Availability Zones und Regionen bei S3 Express One Zone”](#).
- Nach dem Erstellen des Buckets kann der Bucket-Typ nicht mehr geändert werden.

Wählen Sie für Availability Zone eine Availability Zone aus, die für Ihre Computing-Services lokal ist. Eine Liste der Availability Zones, die Directory-Buckets und die Speicherklasse S3 Express One Zone unterstützen, finden Sie unter [the section called “Availability Zones und Regionen bei S3 Express One Zone”](#)

Note

Die Availability Zone kann nicht geändert werden, nachdem der Bucket erstellt wurde.

7. Aktivieren Sie unter Availability Zone das Kontrollkästchen, um zu bestätigen, dass Ihre Daten im Falle eines Ausfalls der Availability Zone möglicherweise nicht verfügbar sind oder verloren gehen.

 **Important**

Obwohl Verzeichnis-Buckets auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert werden, speichern Verzeichnis-Buckets Daten nicht redundant über Availability Zones hinweg.

8. Geben Sie unter Bucket-Name einen Namen für Ihren Verzeichnis-Bucket ein.

Die folgenden Benennungsregeln gelten für Verzeichnis-Buckets.

- Seien Sie innerhalb der Auserwählten einzigartig AWS-Region und Availability Zone.
- Der Name muss zwischen 3 (min) und 63 (maximal) Zeichen lang sein, einschließlich des Suffixes.
- Nur aus Kleinbuchstaben, Zahlen und Bindestrichen bestehen.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.
- Muss das folgende Suffix enthalten: `--azid--x-s3`
- Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
- Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#).
- Bucket-Namen dürfen nicht mit dem Suffix `--o1-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets](#).
- Bucket-Namen dürfen nicht mit dem Suffix `.mrp` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter [Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten](#).

Dem Basisnamen, den Sie angeben, wenn Sie mit der Konsole einen Verzeichnis-Bucket erstellen, wird automatisch ein Suffix hinzugefügt. Dieses Suffix enthält die ID der Availability Zone, die Sie ausgewählt haben.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

 **Important**

Nehmen Sie keine vertraulichen Informationen wie Kontonummern in den Bucket-Namen auf. Der Bucket-Name ist in dem Punkt sichtbarURLs, der auf die Objekte im Bucket verweist.

9. Unter Object Ownership ist die Einstellung Bucket Owner enforced automatisch aktiviert und alle Zugriffskontrolllisten (ACLs) sind deaktiviert. Kann für Verzeichnis-Buckets nicht ACLs aktiviert werden.

ACLsdeaktiviert

- Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

10. Unter „Einstellungen für öffentlichen Zugriff blockieren“ für diesen Bucket werden alle Einstellungen für „Öffentlichen Zugriff blockieren“ für Ihren Verzeichnis-Bucket automatisch aktiviert. Diese Einstellungen können für Directory-Buckets nicht geändert werden. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).
11. Unter Serverseitige Verschlüsselungseinstellungen verwenden Verzeichnis-Buckets serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), um Daten zu verschlüsseln.

12. Wählen Sie Bucket erstellen aus.

Nachdem Sie den Bucket erstellt haben, können Sie ihm Dateien und Ordner hinzufügen. Weitere Informationen finden Sie unter [the section called “Arbeiten mit Objekten in einem Verzeichnis-Bucket”](#).

Verwendung der AWS SDKs

SDK for Go

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie AWS SDK for Go.

Example

```
var bucket = "..."  
  
func runCreateBucket(c *s3.Client) {  
    resp, err := c.CreateBucket(context.Background(), &s3.CreateBucketInput{  
        Bucket: &bucket,  
        CreateBucketConfiguration: &types.CreateBucketConfiguration{  
            Location: &types.LocationInfo{  
                Name: aws.String("usw2-az1"),  
                Type: types.LocationTypeAvailabilityZone,  
            },  
            Bucket: &types.BucketInfo{  
                DataRedundancy: types.DataRedundancySingleAvailabilityZone,  
                Type:           types.BucketTypeDirectory,  
            },  
        },  
    })  
    var terr *types.BucketAlreadyOwnedByYou  
    if errors.As(err, &terr) {  
        fmt.Printf("BucketAlreadyOwnedByYou: %s\n", aws.ToString(terr.Message))  
        fmt.Printf("noop...\n")  
        return  
    }  
    if err != nil {  
        log.Fatal(err)  
    }  
  
    fmt.Printf("bucket created at %s\n", aws.ToString(resp.Location))  
}
```

SDK for Java 2.x

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie AWS SDK for Java 2.x.

Example

```
public static void createBucket(S3Client s3Client, String bucketName) {

    //Bucket name format is {base-bucket-name}--{az-id}--x-s3
    //example: doc-example-bucket--usw2-az1--x-s3 is a valid name for a directory
    bucket created in
    //Region us-west-2, Availability Zone 2

    CreateBucketConfiguration bucketConfiguration =
    CreateBucketConfiguration.builder()
        .location(LocationInfo.builder()
            .type(LocationType.AVAILABILITY_ZONE)
            .name("usw2-az1").build()) //this must match the Region and
    Availability Zone in your bucket name
        .bucket(BucketInfo.builder()
            .type(BucketType.DIRECTORY)
            .dataRedundancy(DataRedundancy.SINGLE_AVAILABILITY_ZONE)
            .build()).build();

    try {

        CreateBucketRequest bucketRequest =
    CreateBucketRequest.builder().bucket(bucketName).createBucketConfiguration(bucketConfiguration)
        CreateBucketResponse response = s3Client.createBucket(bucketRequest);
        System.out.println(response);

    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

}
```

AWS SDK for JavaScript

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie den AWS SDK for JavaScript.

Example

```
// file.mjs, run with Node.js v16 or higher
// To use with the preview build, place this in a folder
// inside the preview build directory, such as /aws-sdk-js-v3/workspace/

import { S3 } from "@aws-sdk/client-s3";

const region = "us-east-1";
const zone = "use1-az4";
const suffix = `${zone}--x-s3`;

const s3 = new S3({ region });

const bucketName = `...--${suffix}`;

const createResponse = await s3.createBucket(
  { Bucket: bucketName,
    CreateBucketConfiguration: {Location: {Type: "AvailabilityZone", Name: zone},
    Bucket: { Type: "Directory", DataRedundancy: "SingleAvailabilityZone" }}
  }
);
```

AWS SDK for .NET

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie den AWS SDK for .NET.

Example

```
using (var amazonS3Client = new AmazonS3Client())
{
    var putBucketResponse = await amazonS3Client.PutBucketAsync(new PutBucketRequest
    {
        BucketName = "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
        PutBucketConfiguration = new PutBucketConfiguration
        {
            BucketInfo = new BucketInfo { DataRedundancy =
            DataRedundancy.SingleAvailabilityZone, Type = BucketType.Directory },
            Location = new LocationInfo { Name = "usw2-az1", Type =
            LocationType.AvailabilityZone }
        }
    });
}
```

```
    }).ConfigureAwait(false);  
}
```

SDK for PHP

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie den AWS SDK for PHP.

Example

```
require 'vendor/autoload.php';  
  
$s3Client = new S3Client([  
    'region' => 'us-east-1',  
]);  
  
$result = $s3Client->createBucket([  
    'Bucket' => 'doc-example-bucket--use1-az4--x-s3',  
    'CreateBucketConfiguration' => [  
        'Location' => ['Name'=> 'use1-az4', 'Type'=> 'AvailabilityZone'],  
        'Bucket' => ["DataRedundancy" => "SingleAvailabilityZone" ,"Type" =>  
            "Directory"]    ],  
    ]);
```

SDK for Python

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie den AWS SDK for Python (Boto3).

Example

```
import logging  
import boto3  
from botocore.exceptions import ClientError  
  
def create_bucket(s3_client, bucket_name, availability_zone):  
    ...  
    Create a directory bucket in a specified Availability Zone  
  
    :param s3_client: boto3 S3 client  
    :param bucket_name: Bucket to create; for example, 'doc-example-bucket--usw2-az1--x-s3'
```

```

    :param availability_zone: String; Availability Zone ID to create the bucket in,
    for example, 'usw2-az1'
    :return: True if bucket is created, else False
    '''

    try:
        bucket_config = {
            'Location': {
                'Type': 'AvailabilityZone',
                'Name': availability_zone
            },
            'Bucket': {
                'Type': 'Directory',
                'DataRedundancy': 'SingleAvailabilityZone'
            }
        }
        s3_client.create_bucket(
            Bucket = bucket_name,
            CreateBucketConfiguration = bucket_config
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    availability_zone = 'usw2-az1'
    s3_client = boto3.client('s3', region_name = region)
    create_bucket(s3_client, bucket_name, availability_zone)

```

SDK for Ruby

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie AWS SDK for Ruby.

Example

```

s3 = Aws::S3::Client.new(region:'us-west-2')
s3.create_bucket(
  bucket: "bucket_base_name--az_id--x-s3",
  create_bucket_configuration: {
    location: { name: 'usw2-az1', type: 'AvailabilityZone' },

```

```
    bucket: { data_redundancy: 'SingleAvailabilityZone', type: 'Directory' }  
  }  
)
```

Verwendung der AWS CLI

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket erstellen, indem Sie den AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

Wenn Sie einen Verzeichnis-Bucket erstellen, müssen Sie Konfigurationsdetails angeben und die folgende Benennungskonvention verwenden: *bucket-base-name--azid--x-s3*

```
aws s3api create-bucket  
--bucket bucket-base-name--azid--x-s3  
--create-bucket-configuration 'Location={Type=AvailabilityZone,Name=usw2-az1},Bucket={DataRedundancy=SingleAvailabilityZone,Type=Directory}'  
--region us-west-2
```

Weitere Informationen finden Sie unter [create-bucket](#) im AWS Command Line Interface.

Anzeigen von Verzeichnis-Bucket-Eigenschaften

Sie können die Eigenschaften für einen Amazon S3 S3-Verzeichnis-Bucket mithilfe der Amazon S3 S3-Konsole anzeigen und konfigurieren. Weitere Informationen erhalten Sie unter [Verzeichnis-Buckets](#) und [Was ist S3 Express One Zone?](#).

Verwenden der S3-Konsole

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie in der Liste Verzeichnis-Buckets den Namen des Buckets aus, dessen Eigenschaften Sie anzeigen möchten.
5. Wählen Sie die Registerkarte Eigenschaften aus.
6. Auf der Registerkarte Eigenschaften können Sie die folgenden Eigenschaften für den Bucket anzeigen:

- Übersicht über den Verzeichnis-Bucket — Sie können Folgendes sehen AWS-Region, Availability Zone, Amazon-Ressourcenname (ARN) und Erstellungsdatum für den Bucket.
- Serverseitige Verschlüsselungseinstellungen — Amazon S3 wendet serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für alle S3-Buckets an. Amazon S3 verschlüsselt ein Objekt, bevor es auf einer Festplatte gespeichert wird, und entschlüsselt das Objekt beim Herunterladen. Weitere Informationen finden Sie unter [Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets](#).

Weitere Informationen zu unterstützten Features für Verzeichnis-Buckets finden Sie unter [Funktionen von S3 Express One Zone](#).

Verwalten von Bucket-Richtlinien für Verzeichnis-Buckets

Mithilfe der Amazon S3-Konsole und der AWS SDKs können Sie Bucket-Richtlinien für Amazon S3 S3-Verzeichnis-Buckets hinzufügen, löschen, aktualisieren und anzeigen. Weitere Informationen finden Sie unter den folgenden Themen. Weitere Informationen zu unterstützten AWS Identity and Access Management (IAM-) Aktionen und Bedingungsschlüsseln für S3 Express One Zone finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#) Beispiele für Bucket-Richtlinien für Verzeichnis-Buckets finden Sie unter [Beispiel für Verzeichnis-Bucket-Richtlinien für S3 Express One Zone](#).

Themen

- [Hinzufügen einer Bucket-Richtlinie](#)
- [Anzeigen einer Bucket-Richtlinie](#)
- [Löschen einer Bucket-Richtlinie](#)

Hinzufügen einer Bucket-Richtlinie

Um eine Bucket-Richtlinie zu einem Verzeichnis-Bucket hinzuzufügen, können Sie die Amazon S3 S3-Konsole, die AWS SDKs oder die AWS CLI verwenden.

Verwenden der S3-Konsole

Eine Bucket-Richtlinie erstellen oder bearbeiten

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie in der Liste Verzeichnis-Buckets den Namen des Buckets aus, in den Ihre Ordner oder Dateien hochgeladen werden sollen.
5. Wählen Sie die Registerkarte Berechtigungen.
6. Wählen Sie unter Bucket-Richtlinie Bearbeiten aus. Die Seite Bucket-Richtlinie bearbeiten wird angezeigt.
7. Um eine Richtlinie automatisch zu generieren, wählen Sie Policy Generator.

Wenn Sie den Richtliniengenerator wählen, wird der AWS Richtliniengenerator in einem neuen Fenster geöffnet.


Wenn Sie den AWS Richtliniengenerator nicht verwenden möchten, können Sie JSON-Anweisungen im Abschnitt Richtlinie hinzufügen oder bearbeiten.

- a. Wählen Sie auf der Seite AWS Policy Generator (Richtliniengenerator) unter Select Type of Policy (Richtlinientyp auswählen) die Option S3 Bucket Policy (S3-Bucket-Richtlinie) aus.
- b. Fügen Sie eine Anweisung hinzu, indem Sie die Informationen in die bereitgestellten Felder eingeben, und wählen Sie dann Anweisung hinzufügen. Wiederholen Sie diesen Schritt für so viele Anweisungen, wie Sie hinzufügen möchten. Weitere Informationen zu diesen Feldern finden Sie in der Referenz zu den [IAM-JSON-Richtlinienelementen](#) im IAM-Benutzerhandbuch.

Note

Der Einfachheit halber wird auf der Seite Bucket-Richtlinie bearbeiten der Bucket-ARN (Amazon-Ressourcenname) des aktuellen Buckets über dem Richtlinientextfeld angezeigt. Sie können diesen ARN zur Verwendung in den Anweisungen auf der Seite AWS -Richtliniengenerator kopieren.

- c. Wenn Sie mit dem Hinzufügen von Anweisungen fertig sind, wählen Sie Generieren von Richtlinien.
 - d. Kopieren Sie den generierten Richtlinienentwurf, wählen Sie Schließen und kehren Sie zur Seite Bucket-Richtlinie bearbeiten in der Amazon-S3-Konsole zurück.
8. Bearbeiten Sie im Feld Richtlinie die bestehende Richtlinie oder fügen Sie die Bucket-Richtlinie aus dem AWS Policy Generator ein. Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge bevor Sie Ihre Richtlinie speichern.

 Note

Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt.

9. Wählen Sie Save changes (Änderungen speichern) aus, wodurch Sie zur Registerkarte Permissions (Berechtigungen) zurückkehren.

Verwenden der AWS SDKs

SDK for Java 2.x

Example

PutBucketPolicy AWS SDK for Java 2.x

```
public static void setBucketPolicy(S3Client s3Client, String bucketName, String
policyText) {

    //sample policy text
    /**
     * policy_statement = {
     *     'Version': '2012-10-17',
     *     'Statement': [
     *         {
     *             'Sid': 'AdminPolicy',
     *             'Effect': 'Allow',
     *             'Principal': {
     *                 "AWS": "111122223333"
     *             },
     *             'Action': 's3express:*',
     *             'Resource':
     * 'arn:aws:s3express:region:111122223333:bucket/bucket-base-name--azid--x-s3'
```

```

*           }
*         ]
*       }
*/
System.out.println("Setting policy:");
System.out.println("----");
System.out.println(policyText);
System.out.println("----");
System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

try {
    PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
        .bucket(bucketName)
        .policy(policyText)
        .build();
    s3Client.putBucketPolicy(policyReq);
    System.out.println("Done!");
}

catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}

```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie Sie mithilfe von eine Bucket-Richtlinie zu einem Verzeichnis-Bucket hinzufügen AWS CLI. Um den Befehl zu verwenden, ersetzen Sie die *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

```
aws s3api put-bucket-policy --bucket bucket-base-name--azid--x-s3 --policy file://
bucket_policy.json
```

bucket_policy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AdminPolicy",
```

```
        "Effect": "Allow",
        "Principal": {
            "AWS": "111122223333"
        },
        "Action": "s3express*",
        "Resource": "arn:aws:s3express:us-west-2:111122223333:bucket/"
    }
]
}
```

Weitere Informationen finden Sie in der [put-bucket-policy](#) AWS Command Line Interface

Anzeigen einer Bucket-Richtlinie

Verwenden Sie die folgenden Beispiele, um eine Bucket-Richtlinie für einen Verzeichnis-Bucket anzuzeigen.

Verwenden von AWS CLI

Dieses Beispiel zeigt, wie die einem Verzeichnis-Bucket zugeordnete Bucket-Richtlinie mithilfe von `aws s3api` angezeigt wird. Um den Befehl zu verwenden, ersetzen Sie die *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

```
aws s3api get-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Weitere Informationen finden Sie [get-bucket-policy](#) in der AWS Command Line Interface.

Löschen einer Bucket-Richtlinie

Verwenden Sie die folgenden Beispiele, um eine Bucket-Richtlinie für einen Verzeichnis-Bucket zu löschen.

Verwenden der AWS SDKs

SDK for Java 2.x

Example

```
DeleteBucketPolicy AWS SDK for Java 2.x
```

```
public static void deleteBucketPolicy(S3Client s3Client, String bucketName) {
    try {
        DeleteBucketPolicyRequest deleteBucketPolicyRequest =
DeleteBucketPolicyRequest
            .builder()
            .bucket(bucketName)
            .build()
        s3Client.deleteBucketPolicy(deleteBucketPolicyRequest);
        System.out.println("Successfully deleted bucket policy");
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie eine Bucket-Richtlinie für einen Verzeichnis-Bucket mithilfe von gelöscht wird AWS CLI. Um den Befehl zu verwenden, ersetzen Sie die *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

```
aws s3api delete-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Weitere Informationen finden Sie [delete-bucket-policy](#) in der AWS Command Line Interface.

Leeren eines Verzeichnis-Buckets


Sie können einen Amazon S3 S3-Verzeichnis-Bucket mithilfe der Amazon S3 S3-Konsole leeren. Weitere Information zu Verzeichnis-Buckets finden Sie unter [Verzeichnis-Buckets](#).

Bevor Sie einen Verzeichnis-Bucket leeren, beachten Sie Folgendes:

- Wenn Sie einen Verzeichnis-Bucket leeren, löschen Sie alle Objekte, behalten aber den Bucket selbst.
- Nachdem Sie einen Verzeichnis-Bucket geleert haben, kann die leere Aktion nicht rückgängig gemacht werden.
- Objekte, die dem Verzeichnis-Bucket hinzugefügt werden, während die Aktion „Leerer Bucket“ ausgeführt wird, werden möglicherweise gelöscht.

Wenn Sie den Bucket auch löschen möchten, beachten Sie Folgendes:

- Alle Objekte in einem Bucket müssen gelöscht werden, bevor der Bucket selbst gelöscht werden kann.
- Laufende mehrteilige Uploads in dem Verzeichnis-Bucket müssen abgebrochen werden, bevor der Bucket selbst gelöscht werden kann.

 Note

Der `s3 rm` Befehl über AWS Command Line Interface (CLI), die `delete` Operation über Mountpoint und die Optionsschaltfläche „Bucket leeren“ über können in Bearbeitung befindliche mehrteilige Uploads in einem Verzeichnis-Bucket nicht löschen. AWS Management Console Um diese laufenden mehrteiligen Uploads zu löschen, verwenden Sie die `ListMultipartUploads` Operation, um die laufenden mehrteiligen Uploads im Bucket aufzulisten und verwenden Sie die Operation, um alle laufenden mehrteiligen Uploads abzuberechnen. `AbortMultipartUpload`

Informationen zum Löschen eines Verzeichnis-Buckets finden Sie unter [Löschen eines Verzeichnis-Buckets](#). Informationen zum Abbrechen eines laufenden mehrteiligen Uploads finden Sie unter [the section called “Abbrechen eines mehrteiligen Uploads”](#)

Informationen zum Leeren eines Allzweck-Buckets finden Sie unter [Leeren eines Buckets](#).

Verwenden der S3-Konsole

So leeren Sie einen Verzeichnis-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie das Optionsfeld neben dem Namen des Buckets, den Sie leeren möchten, und wählen Sie dann Leer.
5. Bestätigen Sie auf der Seite Empty bucket (Bucket leeren), dass Sie den Bucket leeren möchten, indem Sie **permanently delete** in das Textfeld eingeben und dann Empty (Leeren) auswählen.
6. Überwachen Sie den Fortschritt beim Entleeren des Buckets auf der Seite Leerer Bucket: Status.

Löschen eines Verzeichnis-Buckets

Sie können nur leere Amazon S3 S3-Verzeichnis-Buckets löschen. Bevor Sie Ihren Verzeichnis-Bucket löschen, müssen Sie alle Objekte im Bucket löschen und alle laufenden mehrteiligen Uploads abbrechen.

Informationen zum Leeren eines Verzeichnis-Buckets finden Sie unter [Leeren eines Verzeichnis-Buckets](#). Informationen zum Abbrechen eines laufenden mehrteiligen Uploads finden Sie unter [the section called "Abbrechen eines mehrteiligen Uploads"](#)

Informationen zum Löschen eines Allzweck-Buckets finden Sie unter [Löschen eines Buckets](#).

Verwenden der S3-Konsole

Nachdem Sie Ihren Verzeichnis-Bucket geleert und alle laufenden mehrteiligen Uploads abgebrochen haben, können Sie Ihren Bucket löschen.

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie in der Liste der Directory-Buckets die Optionsschaltfläche neben dem Bucket aus, den Sie löschen möchten.
5. Wählen Sie Löschen aus.
6. Geben Sie auf der Seite „Bucket löschen“ den Namen des Buckets in das Textfeld ein, um das Löschen Ihres Buckets zu bestätigen.

Important

Das Löschen eines Verzeichnis-Buckets kann nicht rückgängig gemacht werden.

7. Um Ihren Verzeichnis-Bucket zu löschen, wählen Sie Bucket löschen aus.

Verwenden der AWS SDKs

In den folgenden Beispielen wird ein Verzeichnis-Bucket mithilfe von AWS SDK for Java 2.x und AWS SDK for Python (Boto3) gelöscht.

SDK for Java 2.x

Example

```
public static void deleteBucket(S3Client s3Client, String bucketName) {

    try {
        DeleteBucketRequest del = DeleteBucketRequest.builder()
            .bucket(bucketName)
            .build();
        s3Client.deleteBucket(del);
        System.out.println("Bucket " + bucketName + " has been deleted");
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_bucket(s3_client, bucket_name):
    """
    Delete a directory bucket in a specified Region

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to delete; for example, 'doc-example-bucket--usw2-az1--x-s3'
    :return: True if bucket is deleted, else False
    """

    try:
        s3_client.delete_bucket(Bucket = bucket_name)
    except ClientError as e:
        logging.error(e)
        return False
```



```
    return True

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie ein Verzeichnis-Bucket mithilfe von gelöscht wird AWS CLI. Um den Befehl zu verwenden, ersetzen Sie die *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

```
aws s3api delete-bucket --bucket bucket-base-name--azid--x-s3 --region us-west-2
```

Weitere Informationen finden Sie unter [delete-bucket](#) im. AWS Command Line Interface

Auflisten von Verzeichnis-Buckets

Die folgenden Beispiele zeigen, wie Verzeichnis-Buckets mithilfe der AWS SDKs und AWS der CLI aufgelistet werden.

Verwenden der SDKs AWS

SDK for Java 2.x

Example

Das folgende Beispiel listet Verzeichnis-Buckets mithilfe von auf. AWS SDK for Java 2.x

```
public static void listBuckets(S3Client s3Client) {
    try {
        ListDirectoryBucketsRequest listDirectoryBucketsRequest =
ListDirectoryBucketsRequest.builder().build();
        ListDirectoryBucketsResponse response =
s3Client.listDirectoryBuckets(listDirectoryBucketsRequest);
        if (response.hasBuckets()) {
            for (Bucket bucket: response.buckets()) {
                System.out.println(bucket.name());
            }
        }
    }
}
```

```
        System.out.println(bucket.creationDate());
    }
}

catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

SDK for Python

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for Python (Boto3)

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_directory_buckets(s3_client):
    """
    Prints a list of all directory buckets in a Region

    :param s3_client: boto3 S3 client
    :return: True if there are buckets in the Region, else False
    """
    try:
        response = s3_client.list_directory_buckets()
        for bucket in response['Buckets']:
            print (bucket['Name'])
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    region = 'us-east-1'
    s3_client = boto3.client('s3', region_name = region)
```

```
list_directory_buckets(s3_client)
```

AWS SDK for .NET

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for .NET

```
var listDirectoryBuckets = await amazonS3Client.ListDirectoryBucketsAsync(new
    ListDirectoryBucketsRequest
{
    MaxDirectoryBuckets = 10
}).ConfigureAwait(false);
```

SDK for PHP

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for PHP

```
require 'vendor/autoload.php';

$s3Client = new S3Client([
    'region'      => 'us-east-1',
]);
$result = $s3Client->listDirectoryBuckets();
```

SDK for Ruby

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for Ruby

```
s3 = Aws::S3::Client.new(region: 'us-west-1')
s3.list_directory_buckets
```

Unter Verwendung der AWS CLI

Der folgende `list-directory-buckets` Beispielbefehl zeigt, wie Sie den verwenden können AWS CLI , um Ihre Verzeichnis-Buckets in der Region `us-east-1` aufzulisten. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api list-directory-buckets --region us-east-1
```

Weitere Informationen finden Sie unter [list-directory-buckets](#) in der Referenz zum AWS CLI -Befehl.

Verwendung **HeadBucket** mit Verzeichnis-Buckets

Die folgenden AWS SDK-Beispiele zeigen, wie Sie mithilfe der `HeadBucket` API-Operation feststellen können, ob ein Amazon S3 S3-Verzeichnis-Bucket vorhanden ist und ob Sie über die Zugriffsberechtigung verfügen.

Verwendung der AWS SDKs

Das folgende AWS SDK for Java 2.x Beispiel zeigt, wie Sie feststellen können, ob ein Bucket vorhanden ist und ob Sie die Berechtigung haben, darauf zuzugreifen.

SDK for Java 2.x

Example

AWS SDK for Java 2.x

```
public static void headBucket(S3Client s3Client, String bucketName) {
    try {
        HeadBucketRequest headBucketRequest = HeadBucketRequest
            .builder()
            .bucket(bucketName)
            .build();
        s3Client.headBucket(headBucketRequest);
        System.out.format("Amazon S3 bucket: \"%s\" found.", bucketName);
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

```
}  
}
```

Mit dem AWS CLI

Der folgende `head-bucket` Beispielbefehl zeigt, wie Sie mit dem ermitteln können AWS CLI, ob ein Verzeichnis-Bucket vorhanden ist und ob Sie über die erforderlichen Zugriffsrechte verfügen. Um diesen Befehl auszuführen, ersetzen Sie die Platzhalter für Benutzereingaben durch Ihre eigenen Informationen.

```
aws s3api head-bucket --bucket bucket-base-name--azid--x-s3
```

Weitere Informationen finden Sie unter [head-bucket](#) in der Referenz zum AWS CLI -Befehl.

Arbeiten mit Objekten in einem Verzeichnis-Bucket

Nachdem Sie einen Amazon S3 S3-Verzeichnis-Bucket erstellt haben, können Sie mithilfe der Amazon S3 S3-Konsole AWS Command Line Interface (AWS CLI) und der AWS SDKs mit Objekten arbeiten.

Weitere Informationen zu Massenoperationen mit Objekten, die in der Speicherklasse S3 Express One Zone gespeichert sind, finden Sie unter [Verwaltung von Objekten](#). Weitere Informationen zum Importieren, Hochladen, Kopieren, Löschen und Herunterladen von Objekten und zum Lesen von Metadaten aus Objekten in Verzeichnis-Buckets finden Sie in den folgenden Themen.

Themen

- [Importieren von Objekten in einen Verzeichnis-Bucket](#)
- [Verwenden von Batch Operations mit S3 Express One Zone](#)
- [Hochladen eines Objekts zu einem Verzeichnis-Bucket](#)
- [Mehrteilige Uploads mit Verzeichnis-Buckets verwenden](#)
- [Hochladen eines Objekts zu einem Verzeichnis-Bucket](#)
- [Löschen eines Objekts in einem Verzeichnis-Bucket](#)
- [Ein Objekt in einem Verzeichnis-Bucket herunterladen](#)
- [Verwendung HeadObject mit Verzeichnis-Buckets](#)

Importieren von Objekten in einen Verzeichnis-Bucket

Nachdem Sie einen Verzeichnis-Bucket in Amazon S3 erstellt haben, können Sie den neuen Bucket mithilfe der Importaktion mit Daten füllen. Der Import ist eine optimierte Methode zur Erstellung von S3-Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets.

Note

Die folgenden Einschränkungen gelten für Importaufträge:

- Der Quell-Bucket und der Ziel-Bucket müssen sich im selben befinden AWS-Region und Konto.
- Der Quell-Bucket kann kein Verzeichnis-Bucket sein.
- Objekte, die größer als 5 GB sind, werden nicht unterstützt und beim Kopiervorgang nicht berücksichtigt.
- Objekte der Speicherklassen Glacier Flexible Retrieval, Glacier Deep Archive, Intelligent-Tiering Archive Access und Intelligent-Tiering Deep Archive müssen wiederhergestellt werden, bevor sie importiert werden können.
- Importierte Objekte mit MD5 Prüfsummenalgorithmen werden so konvertiert, dass sie CRC32 Prüfsummen verwenden.
- Importierte Objekte verwenden die Speicherklasse Express One Zone, die eine andere Preisstruktur hat als die Speicherklassen, die in Buckets für allgemeine Zwecke verwendet werden. Beachten Sie diesen Kostenunterschied, wenn Sie eine große Anzahl von Objekten importieren.

Wenn Sie einen Importauftrag konfigurieren, geben Sie den Quell-Bucket oder das Quellpräfix an, aus dem die vorhandenen Objekte kopiert werden. Sie stellen auch eine AWS Identity and Access Management (IAM) Rolle, die über Berechtigungen für den Zugriff auf die Quellobjekte verfügt. Amazon S3 startet dann einen Batch-Operations-Auftrag, der die Objekte kopiert und automatisch die entsprechenden Speicherklassen- und Prüfsummeneinstellungen anwendet.

Um Importaufträge zu konfigurieren, verwenden Sie die Amazon-S3-Konsole.

Verwenden der Amazon S3-Konsole

So importieren Sie Objekte in einen Verzeichnis-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets und dann die Registerkarte Verzeichnis-Buckets aus. Wählen Sie die Optionsschaltfläche neben dem Verzeichnis-Bucket, in den Sie Objekte importieren möchten.
3. Wählen Sie Importieren aus.
4. Geben Sie unter Quelle den Allzweck-Bucket (oder den Bucket-Pfad einschließlich Präfix) ein, der die Objekte enthält, die Sie importieren möchten. Um einen vorhandenen Allzweck-Bucket aus einer Liste auszuwählen, wählen Sie S3 durchsuchen.
5. Gehen Sie für die Berechtigung zum Zugreifen auf und Kopieren von Quellobjekten auf eine der folgenden Arten vor, um eine IAM Rolle mit den erforderlichen Berechtigungen für den Import Ihrer Quellobjekte anzugeben:
 - Um Amazon S3 zu ermöglichen, in Ihrem Namen eine neue IAM Rolle zu erstellen, wählen Sie Neue IAM Rolle erstellen.
 - Um eine bestehende IAM Rolle aus einer Liste auszuwählen, wählen Sie Aus vorhandenen IAM Rollen auswählen.
 - Um eine bestehende IAM Rolle durch Eingabe ihres Amazon-Ressourcennamens (ARN) anzugeben, wählen Sie IAM Rolle eingeben ARN und geben Sie dann das ARN in das entsprechende Feld ein.
6. Überprüfen Sie die Informationen, die in den Abschnitten Ziel und Kopierte Objekteinstellungen angezeigt werden. Wenn die Informationen im Bereich Ziel korrekt sind, wählen Sie Import, um den Kopierauftrag zu starten.

Die Amazon-S3-Konsole zeigt den Status Ihres neuen Auftrags auf der Seite Batch-Operationen an. Für weitere Informationen über den Auftrag klicken Sie auf das Optionsfeld neben dem Auftragsnamen und wählen dann im Menü Aktionen die Option Details anzeigen. Um den Verzeichnis-Bucket zu öffnen, in den die Objekte importiert werden, wählen Sie Importziel anzeigen.

Verwenden von Batch Operations mit S3 Express One Zone

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Vorgänge für in S3 gespeicherte Objekte durchzuführen. Weitere Informationen zu S3 Batch Operations finden Sie unter [Durchführen umfangreicher Stapelvorgänge für Amazon-S3-Objekte](#).

In den folgenden Themen wird die Durchführung von Batch-Vorgängen für Objekte beschrieben, die in der Speicherklasse S3 Express One Zone in Verzeichnis-Buckets gespeichert sind.

Themen

- [Verwenden von Batch Operations mit Verzeichnis-Buckets](#).
- [Die wichtigsten Unterschiede](#):

Verwenden von Batch Operations mit Verzeichnis-Buckets.

Sie können den Kopiervorgang und den Aufrufvorgang ausführen AWS Lambda Funktionsoperationen für Objekte, die in Verzeichnis-Buckets gespeichert sind. Mit Copy können Sie Objekte zwischen Buckets desselben Typs kopieren (z. B. von einem Verzeichnis-Bucket in einen Verzeichnis-Bucket). Sie können auch zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Mit Invoke AWS Lambda Funktion, Sie können eine Lambda-Funktion verwenden, um Aktionen an Objekten in Ihrem Verzeichnis-Bucket mit von Ihnen definiertem Code auszuführen.

Objekte kopieren

Sie können zwischen demselben Bucket-Typ oder zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Wenn Sie in einen Verzeichnis-Bucket kopieren, müssen Sie das richtige Amazon-Ressourcenname (ARN) -Format für diesen Bucket-Typ verwenden. Das ARN Format für einen Verzeichnis-Bucket ist `arn:aws:s3express:region:account-id:bucket/bucket-base-name--x-s3`.

Sie können Ihren Verzeichnis-Bucket auch mit Daten füllen, indem Sie die Aktion Import in der S3-Konsole verwenden. Import ist eine optimierte Methode zur Erstellung von Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets. Für Import-Kopieraufträge aus Allzweck-Buckets in Verzeichnis-Buckets generiert S3 automatisch ein Manifest. Weitere Informationen finden Sie unter [Objekte in einen Verzeichnis-Bucket importieren](#) und [Ein Manifest angeben](#).

Lambda-Funktionen () aufrufen **LambdaInvoke**

Für die Verwendung von Batch Operations zum Aufrufen von Lambda-Funktionen, die auf Verzeichnis-Buckets wirken, gelten besondere Anforderungen. Beispielsweise müssen Sie Ihre Lambda-Anfrage strukturieren, indem Sie ein v2 JSONAufrufschema und spezifizieren InvocationSchemaVersion 2.0 wenn Sie den Job erstellen. Weitere Informationen finden Sie unter [Aufrufen AWS Lambda Funktion](#).

Die wichtigsten Unterschiede:

Im Folgenden finden Sie eine Liste der wichtigsten Unterschiede, wenn Sie Batch Operations verwenden, um Massenoperationen an Objekten durchzuführen, die in Verzeichnis-Buckets mit der S3 Express One Zone-Speicherkategorie gespeichert sind:

- Für Verzeichnis-Buckets, SSE -S3 und serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) werden unterstützt. Wenn Sie eine CopyObject Anfrage stellen, die angibt, serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) für einen Verzeichnis-Bucket (Quelle oder Ziel) zu verwenden, gibt die Antwort einen Fehler zurück. HTTP 400 (Bad Request)

Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren CreateSession Anfragen oder PUT Objektanforderungen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets und zum Verschlüsseln neuer Objektkopien in einem Verzeichnis-Bucket mit SSE - finden Sie unter [KMS Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).

S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte aus Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE, und zwar über [Copy Betrieb in Batch-Operationen](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS - verschlüsseltes Objekt gestellt wird. Weitere Hinweise zur Verwendung von SSE - in KMS Verzeichnis-Buckets finden Sie unter [Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets](#) und [Serverseitige Verschlüsselung verwenden mit AWS KMS Schlüssel \(SSE-KMS\) in Verzeichnis-Buckets](#)

- Objekte in Verzeichnis-Buckets können nicht mit Tags markiert werden. Sie können nur einen leeren Tag-Satz angeben. Standardmäßig kopiert Batch Operations Tags. Wenn Sie ein Objekt

mit Tags zwischen Allzweck-Buckets und Verzeichnis-Buckets kopieren, erhalten Sie eine Antwort. 501 (Not Implemented)

- S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten bei Uploads oder Downloads verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) wählen: CRC32, SHA -1 und -256. SHA MD5basierte Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.
- Standardmäßig setzen alle Amazon S3 S3-Buckets die Einstellung S3 Object Ownership auf Bucket Owner enforced und die Zugriffskontrolllisten (ACLs) sind deaktiviert. Für Verzeichnis-Buckets kann diese Einstellung nicht geändert werden. Sie können ein Objekt aus Allzweck-Buckets in Verzeichnis-Bucket kopieren. Sie können die Standardeinstellung jedoch nicht überschreiben, ACL wenn Sie in oder aus einem Verzeichnis-Bucket kopieren.
- Unabhängig davon, wie Sie Ihr Manifest angeben, muss die Liste selbst in einem Allzweck-Bucket gespeichert werden. Batch Operations kann keine vorhandenen Manifeste aus Verzeichnis-Buckets importieren (oder generierte Manifeste in Verzeichnis-Buckets speichern). Im Manifest beschriebene Objekte können jedoch in Verzeichnis-Buckets gespeichert werden.
- Batch Operations kann keinen Verzeichnis-Bucket als Speicherort in einem S3-Inventarbericht angeben. Inventarberichte unterstützen keine Verzeichnis-Buckets. Sie können eine Manifestdatei für Objekte innerhalb eines Verzeichnis-Buckets erstellen, indem Sie den ListObjectsV2 API Vorgang zum Auflisten der Objekte verwenden. Anschließend können Sie die Liste in eine CSV Datei einfügen.

Gewähren von -Zugriff

Um Kopieraufträge durchzuführen müssen Sie über die folgenden Berechtigungen verfügen:

- Um Objekte von einem Verzeichnis-Bucket zu einem anderen Verzeichnis-Bucket zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung.
- Um Objekte aus Verzeichnis-Buckets zu Allzweck-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung und `s3:PutObject`-Berechtigung zum Schreiben der Objektkopie in den Ziel-Bucket.
- Um Objekte aus Allzweck-Buckets in Verzeichnis-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession` Berechtigung und die Berechtigung, das `s3:GetObject` kopierte Quellobjekt zu lesen.

Weitere Informationen finden Sie unter [CopyObject](#) in der Amazon Simple Storage Service API Reference.

- Um eine Lambda-Funktion aufzurufen, müssen Sie Ihrer Ressource basierend auf Ihrer Lambda-Funktion Berechtigungen erteilen. Um festzustellen, welche Berechtigungen erforderlich sind, überprüfen Sie die entsprechenden API Vorgänge.

Hochladen eines Objekts zu einem Verzeichnis-Bucket

Nachdem Sie einen Amazon S3-Verzeichnis-Bucket erstellt haben, können Sie Objekte in diesen hochladen. Die folgenden Beispiele zeigen, wie Sie mithilfe der S3-Konsole und der AWS SDKs ein Objekt in einen Verzeichnis-Bucket hochladen. Informationen zum Hochladen von Objekten in großen Mengen mit S3 Express One Zone finden Sie unter [Verwaltung von Objekten](#).

Verwenden der S3-Konsole

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie den Namen des Buckets, in den Sie Ihre Ordner oder Dateien hochladen möchten.
5. Wählen Sie in der Objektliste die Option Hochladen aus.
6. Führen Sie auf der Upload-Seite einen der folgenden Schritte aus:
 - Ziehen Sie Dateien und Ordner per Drag & Drop in den gepunkteten Upload-Bereich.
 - Wählen Sie „Dateien hinzufügen“ oder „Ordner hinzufügen“, wählen Sie die Dateien oder Ordner aus, die Sie hochladen möchten, und wählen Sie dann „Öffnen“ oder „Hochladen“.
7. Wählen Sie unter Prüfsummen die Prüfsummenfunktion aus, die Sie verwenden möchten.

(Optional) Wenn Sie ein einzelnes Objekt hochladen, das weniger als 16 MB groß ist, können Sie auch einen vorab berechneten Prüfsummenwert angeben. Wenn Sie einen vorberechneten Wert angeben, vergleicht Amazon S3 ihn mit dem Wert, den es mithilfe der ausgewählten Prüfsummenfunktion berechnet. Wenn die Werte nicht übereinstimmen, startet der Upload nicht.

8. Die Optionen in den Abschnitten „Berechtigungen“ und „Eigenschaften“ werden automatisch auf die Standardeinstellungen gesetzt und können nicht geändert werden. Block Public Access ist

automatisch aktiviert, und S3 Versioning und S3 Object Lock können für Directory-Buckets nicht aktiviert werden.

(Optional) Wenn Sie Ihren Objekten Metadaten in Schlüssel-Wert-Paaren hinzufügen möchten, erweitern Sie den Abschnitt Eigenschaften und wählen Sie dann im Abschnitt Metadaten die Option Metadaten hinzufügen aus.

9. Um die aufgelisteten Dateien und Ordner hochzuladen, wählen Sie Hochladen.

Amazon S3 lädt Ihre Objekte und Ordner hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: Status eine Erfolgsmeldung angezeigt.

Verwenden der AWS SDKs

SDK for Java 2.x

Example

```
public static void putObject(S3Client s3Client, String bucketName, String objectKey,
    Path filePath) {
    //Using File Path to avoid loading the whole file into memory
    try {
        PutObjectRequest putObj = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            //.metadata(metadata)
            .build();
        s3Client.putObject(putObj, filePath);
        System.out.println("Successfully placed " + objectKey + " into bucket
"+bucketName);

    }

    catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

```
import boto3
import botocore
from botocore.exceptions import ClientError

def put_object(s3_client, bucket_name, key_name, object_bytes):
    """
    Upload data to a directory bucket.
    :param s3_client: The boto3 S3 client
    :param bucket_name: The bucket that will contain the object
    :param key_name: The key of the object to be uploaded
    :param object_bytes: The data to upload
    """
    try:
        response = s3_client.put_object(Bucket=bucket_name, Key=key_name,
                                         Body=object_bytes)
        print(f"Upload object '{key_name}' to bucket '{bucket_name}'.")
        return response
    except ClientError:
        print(f"Couldn't upload object '{key_name}' to bucket '{bucket_name}'.")
        raise

def main():
    # Share the client session with functions and objects to benefit from S3 Express
    # One Zone auth key
    s3_client = boto3.client('s3')
    # Directory bucket name must end with --azid--x-s3
    resp = put_object(s3_client, 'doc-bucket-example--use1-az5--x-s3', 'sample.txt',
                      b'Hello, World!')
    print(resp)

if __name__ == "__main__":
    main()
```

Mit dem AWS CLI

Der folgende `put-object` Beispielfehl zeigt, wie Sie mit AWS CLI dem ein Objekt aus Amazon S3 hochladen können. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api put-object --bucket bucket-base-name--azid--x-s3 --key sampleinput/file001.bin
--body bucket-seed/file001.bin
```

Weitere Informationen finden Sie unter [put-object](#) in der Referenz zum AWS CLI -Befehl.

Mehrteilige Uploads mit Verzeichnis-Buckets verwenden

Sie können den mehrteiligen Upload-Prozess verwenden, um ein einzelnes Objekt als Satz von Teilen hochzuladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt. Wenn Ihre Objektgröße 100 MB erreicht, sollten Sie in der Regel mehrteilige Uploads verwenden, anstatt das Objekt in einem einzigen Vorgang hochzuladen.

Die Nutzung mehrteiliger Uploads bietet die folgenden Vorteile:

- Verbesserter Durchsatz – Sie können die Teile parallel hochladen, um den Durchsatz zu erhöhen.
- Schnelle Wiederherstellung bei Netzwerkproblemen — Kleinere Bauteile minimieren die Auswirkungen eines Neustarts eines aufgrund eines Netzwerkfehlers fehlgeschlagenen Uploads.
- Anhalten und Fortsetzen von Objekt-Uploads – Sie können Objektteile mit der Zeit hochladen. Nachdem Sie einen mehrteiligen Upload initiiert haben, gibt es kein Ablaufdatum. Sie müssen den mehrteiligen Upload explizit abschließen oder abbrechen.
- Starten Sie einen Upload, bevor Sie die endgültige Objektgröße kennen – Sie können ein Objekt hochladen, während Sie es noch erstellen.

Wir empfehlen, mehrteilige Uploads auf folgende Weise zu verwenden:

- Wenn Sie große Objekte über ein stabiles Netzwerk mit hoher Bandbreite hochladen, verwenden Sie mehrteilige Uploads, um die verfügbare Bandbreite optimal zu nutzen, indem Sie Objektteile parallel hochladen, um eine Multithread-Leistung zu erzielen.

- Wenn Sie über ein lückenhaftes Netzwerk hochladen, verwenden Sie mehrteilige Uploads, um die Widerstandsfähigkeit gegenüber Netzwerkfehlern zu erhöhen, indem Sie Neustarts des Uploads vermeiden. Wenn Sie mehrteilige Uploads verwenden, müssen Sie erneut versuchen, nur die Teile hochzuladen, die während des Uploads unterbrochen wurden. Sie müssen nicht das gesamte Objekt von Anfang an neu hochladen.

Wenn Sie mehrteilige Uploads verwenden, um Objekte in Verzeichnis-Buckets in die Amazon S3 Express One Zone-Speicherklasse hochzuladen, ähnelt der mehrteilige Upload-Vorgang dem Vorgang des mehrteiligen Uploads zum Hochladen von Objekten in Allzweck-Buckets. Es gibt jedoch einige grundlegende Unterschiede.

Weitere Informationen zur Verwendung von mehrteiligen Uploads zum Hochladen von Objekten in die S3 Express One Zone finden Sie in den folgenden Themen.

Themen

- [Der mehrteilige Upload-Vorgang](#)
- [Prüfsummen mit mehrteiligen Upload-Operationen](#)
- [Gleichzeitige mehrteilige Upload-Vorgänge](#)
- [Mehrteilige Uploads und Preisgestaltung](#)
- [Operationen und Berechtigungen für mehrteilige Uploads API](#)
- [Beispiele](#)

Der mehrteilige Upload-Vorgang

Ein mehrteiliger Upload besteht aus drei Schritten:

- Sie initiieren den Upload.
- Sie laden die Objektteile hoch.
- Nachdem Sie alle Teile hochgeladen haben, schließen Sie den mehrteiligen Upload ab.

Nach Erhalt der vollständigen mehrteiligen Upload-Anfrage erstellt Amazon S3 das Objekt aus den hochgeladenen Teilen, und Sie können dann wie auf jedes andere Objekt in Ihrem Bucket auf das Objekt zugreifen.

Initiieren des mehrteiligen Uploads

Wenn Sie eine Anfrage zum Initiieren eines mehrteiligen Uploads senden, gibt Amazon S3 eine Antwort mit einer Upload-ID zurück, als eindeutige Kennung für den Multipart-Upload. Sie müssen diese Upload-ID immer angeben, wenn Sie Teile hochladen, die Teile auflisten, einen Upload durchführen oder ihn abbrechen.

Teile hochladen

Beim Hochladen eines Teils müssen Sie zusätzlich zur Upload-ID eine Teilenummer angeben. Wenn Sie einen mehrteiligen Upload mit S3 Express One Zone verwenden, müssen die mehrteiligen Artikelnummern fortlaufende Artikelnummern sein. Wenn Sie versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, wird ein Fehler HTTP 400 Bad Request (Ungültige Bestellung) generiert.

Eine Artikelnummer identifiziert eindeutig ein Teil und seine Position in dem Objekt, das Sie hochladen. Wenn Sie ein neues Teil hochladen und dabei dieselbe Artikelnummer wie ein zuvor hochgeladenes Teil verwenden, wird das zuvor hochgeladene Teil überschrieben.

Immer wenn Sie ein Teil hochladen, gibt Amazon S3 in seiner Antwort einen Entity-Tag (ETag) - Header zurück. Für jeden Upload eines Teils müssen Sie die Artikelnummer und den ETag Wert aufzeichnen. Die ETag Werte für alle Uploads von Objektteilen bleiben gleich, aber jedem Teil wird eine andere Artikelnummer zugewiesen. Sie müssen diese Werte in die spätere Anforderung einschließen, um den mehrteiligen Upload abzuschließen.

Amazon S3 verschlüsselt automatisch alle neuen Objekte, die in einen S3-Bucket hochgeladen werden. Wenn Sie bei einem mehrteiligen Upload in Ihrer Kopieranforderung keine Verschlüsselungsinformationen angeben, wird die Verschlüsselungseinstellung der hochgeladenen Teile auf die Standardverschlüsselungskonfiguration des Ziel-Buckets festgelegt. Die Standardverschlüsselungskonfiguration eines Amazon S3 S3-Buckets ist immer aktiviert und mindestens auf serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) eingestellt. Für Verzeichnis-Buckets, SSE -S3 und serverseitige Verschlüsselung mit AWS KMS Schlüssel (SSE-KMS) werden unterstützt. Weitere Informationen finden Sie unter [Datenschutz und Verschlüsselung in S3 Express One Zone](#).

Abschließen eines mehrteiligen Uploads

Wenn Sie einen mehrteiligen Upload abschließen, erstellt Amazon S3 das Objekt, indem es die Teile anhand der Artikelnummer in aufsteigender Reihenfolge verkettet. Nach einer erfolgreich ausgeführten Abschlussanforderung sind die Teile nicht mehr vorhanden.

Ihre vollständige mehrteilige Upload-Anfrage muss die Upload-ID und eine Liste der beiden Artikelnummern und der entsprechenden Werte enthalten. ETag Die Amazon S3 S3-Antwort enthält eineETag, die die kombinierten Objektdaten eindeutig identifiziert. Dies ETag ist kein MD5 Hash der Objektdaten.

Auflistungen mehrteiliger Uploads

Sie können alle Teile eines bestimmten Multipart-Uploads oder alle laufenden mehrteiligen Uploads auflisten. Die Operation für die Teileauflistung gibt die Teileinformationen zurück, die Sie für einen bestimmten mehrteiligen Upload hochgeladen haben. Für jeden Abruf einer Teileauflistung gibt Amazon S3 die Teileinformationen für einen angegebenen mehrteiligen Upload bis zu maximal 1.000 Teilen zurück. Wenn in dem mehrteiligen Upload mehr als 1 000 Teile vorhanden sind, müssen Sie die Paginierung verwenden, um alle Teile abzurufen.

Die zurückgegebene Teileliste enthält keine Teile, deren Upload noch nicht abgeschlossen ist. Bei Verwendung der Operation Mehrteilige Uploads auflisten können Sie eine Liste aller mehrteiligen Uploads erhalten, die sich in Bearbeitung befinden.

Ein laufender mehrteiliger Upload ist ein Upload, den Sie initiiert haben, der aber noch nicht abgeschlossen ist oder abgebrochen wurde. Jeder Anforderung gibt bis zu 1.000 mehrteilige Uploads zurück. Wenn mehr als 1 000 mehrteilige Uploads vorhanden sind, müssen Sie zusätzliche Anforderungen senden, um die verbleibenden mehrteiligen Uploads abzurufen. Verwenden Sie die zurückgegebene Liste nur zur Überprüfung. Verwenden Sie das Ergebnis dieser Auflistung nicht, wenn Sie eine Anforderung für den Abschluss eines mehrteiligen Uploads senden. Pflegen Sie stattdessen Ihre eigene Liste mit den Artikelnummern, die Sie beim Hochladen von Teilen angegeben haben, und den entsprechenden ETag Werten, die Amazon S3 zurückgibt.

Weitere Informationen zu mehrteiligen Upload-Angeboten finden Sie unter [ListParts](#) in der Amazon Simple Storage Service API Reference.

Prüfsummen mit mehrteiligen Upload-Operationen

Wenn Sie ein Objekt hochladen, können Sie einen Prüfsummenalgorithmus angeben, um die Objektintegrität zu überprüfen. MD5 wird für Verzeichnis-Buckets nicht unterstützt. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität mit Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) angeben:

- CRC32
- CRC32C

- SHA-1
- SHA-256

Sie können den Amazon S3 REST API oder den AWS SDKsum den Prüfsummenwert für einzelne Teile mithilfe von `GetObject` oder `HeadObject` abzurufen. Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie `ListParts` verwenden.

Important

Wenn Sie die oben genannten Prüfsummenalgorithmen verwenden, müssen die mehrteiligen Artikelnummern fortlaufende Artikelnummern verwenden. Wenn Sie versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen Fehler HTTP 400 Bad Request (Ungültige Teilbestellung).

Weitere Informationen zur Funktionsweise von Prüfsummen mit mehrteiligen Objekten finden Sie unter [Überprüfung der Objektintegrität](#).

Gleichzeitige mehrteilige Upload-Vorgänge

In einer verteilten Entwicklungsumgebung kann Ihre Anwendung mehrere Aktualisierungen für dasselbe Objekt gleichzeitig initiieren. Beispielsweise könnte Ihre Anwendung mehrere mehrteilige Uploads mithilfe desselben Objektschlüssels initiieren. Für jeden dieser Uploads kann Ihre Anwendung Teile hochladen und eine Anfrage auf Abschluss des Uploads an Amazon S3 senden, um das Objekt zu erstellen. Für S3 Express One Zone entspricht die Objekterstellungszeit dem Abschlussdatum des mehrteiligen Uploads.

Important

Die Versionierung wird für Objekte, die in Verzeichnis-Buckets gespeichert sind, nicht unterstützt.

Mehrteilige Uploads und Preisgestaltung

Nachdem Sie einen mehrteiligen Upload gestartet haben, behält Amazon S3 alle Teile bei, bis Sie den Upload abschließen oder abbrechen. Während seiner gesamten Lebensdauer werden

Ihnen der gesamte Speicher, die Bandbreite und die Anforderungen für diesen mehrteiligen Upload und die zugehörigen Teile in Rechnung gestellt. Wenn Sie den mehrteiligen Upload abbrechen, löscht Amazon S3 die Upload-Artefakte und alle Teile, die Sie hochgeladen haben, und sie werden Ihnen nicht mehr in Rechnung gestellt. Für das Löschen unvollständiger mehrteiliger Uploads fallen unabhängig von der angegebenen Speicherklasse keine Gebühren für vorzeitiges Löschen an. Weitere Informationen zu Preisen finden Sie unter [Amazon-S3-Preise](#).

Important

Wenn die vollständige mehrteilige Upload-Anfrage nicht erfolgreich gesendet wird, werden die Objektteile nicht zusammengesetzt und es wird kein Objekt erstellt. Ihnen wird der gesamte Speicher in Rechnung gestellt, der mit hochgeladenen Teilen verknüpft ist. Es ist wichtig, dass Sie entweder den mehrteiligen Upload abschließen, damit das Objekt erstellt wird, oder den mehrteiligen Upload abbrechen, um alle hochgeladenen Teile zu entfernen.

Bevor Sie einen Verzeichnis-Bucket löschen können, müssen Sie alle laufenden mehrteiligen Uploads abschließen oder abbrechen. Directory-Buckets unterstützen keine S3-Lifecycle-Konfigurationen. Bei Bedarf können Sie Ihre aktiven mehrteiligen Uploads auflisten, die Uploads dann abbrechen und dann Ihren Bucket löschen.

Operationen und Berechtigungen für mehrteilige Uploads API

Um den Zugriff auf API Objektverwaltungsvorgänge in einem Verzeichnis-Bucket zu ermöglichen, erteilen Sie die `s3express:CreateSession` Erlaubnis in einer Bucket-Richtlinie oder in AWS Identity and Access Management (IAM) identitätsbasierte Richtlinie.

Sie müssen über die erforderlichen Berechtigungen verfügen, um die Multipart-Upload-Vorgänge zu verwenden. Sie können Bucket-Richtlinien oder IAM identitätsbasierte Richtlinien verwenden, um IAM Prinzipalen Berechtigungen zur Ausführung dieser Operationen zu gewähren. Die folgende Tabelle listet die erforderlichen Berechtigungen für verschiedene mehrteilige Uploadvorgänge auf.

Sie können den Initiator eines mehrteiligen Uploads anhand des Elements identifizieren. `Initiator` Wenn der Initiator ein ist AWS-Konto, dieses Element liefert dieselben Informationen wie das `Owner` Element. Wenn der Initiator ein IAM Benutzer ist, stellt dieses Element den Benutzer ARN und den Anzeigenamen bereit.

Aktion	Erforderliche Berechtigungen
Erstellen eines mehrteiligen Uploads	Um den mehrteiligen Upload zu erstellen, müssen Sie berechtigt sein, die <code>s3express:CreateSession</code> Aktion im Verzeichnis-Bucket auszuführen.
Initiieren Sie einen mehrteiligen Upload	Um den mehrteiligen Upload zu initiieren, müssen Sie berechtigt sein, die <code>s3express:CreateSession</code> Aktion für den Verzeichnis-Bucket auszuführen.
Laden Sie einen Teil hoch	Um ein Teil hochzuladen, müssen Sie berechtigt sein, die <code>s3express:CreateSession</code> Aktion im Verzeichnis-Bucket auszuführen. Damit der Initiator einen Teil hochladen kann, muss der Bucket-Besitzer dem Initiator erlauben, die <code>s3express:CreateSession</code> Aktion für den Verzeichnis-Bucket auszuführen.
Laden Sie einen Teil hoch (Kopie)	Um ein Teil hochzuladen, müssen Sie berechtigt sein, die <code>s3express:CreateSession</code> Aktion im Verzeichnis-Bucket auszuführen. Damit der Initiator einen Teil eines Objekts hochladen kann, muss der Bucket-Besitzer dem Initiator eine Genehmigung für die Ausführung der Aktion <code>s3express:CreateSession</code> für das Objekt erteilen.
Abschließen eines mehrteiligen Uploads	Um einen mehrteiligen Upload abzuschließen, müssen Sie berechtigt sein, die <code>s3express:CreateSession</code> Aktion im Verzeichnis-Bucket auszuführen. Damit der Initiator einen mehrteiligen Upload abschließen kann, muss der Bucket-Besitzer dem Initiator gestatten, die <code>s3express:CreateSession</code> Aktion für das Objekt auszuführen.
Abbrechen eines mehrteiligen Uploads	Um einen mehrteiligen Upload abbrechen, müssen Sie berechtigt sein, die Aktion auszuführen. <code>s3express:CreateSession</code> Damit der Initiator einen mehrteiligen Upload abbrechen kann, muss dem Initiator die ausdrückliche Zugriffsberechtigung für die Ausführung der Aktion gewährt werden. <code>s3express:CreateSession</code>

Aktion	Erforderliche Berechtigungen
Teile auflisten	Um die Teile in einem mehrteiligen Upload aufzulisten, müssen Sie berechtigt sein, die <code>s3express:CreateSession</code> Aktion im Verzeichnis-Bucket auszuführen.
Laufende mehrteilige Uploads auflisten	Um die laufenden mehrteiligen Uploads in einen Bucket aufzulisten, müssen Sie berechtigt sein, die <code>s3:ListBucketMultipartUploads</code> Aktion für diesen Bucket auszuführen.

API-Betriebsunterstützung für mehrteilige Uploads

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference werden die Amazon S3 REST API S3-Operationen für mehrteilige Uploads beschrieben.

- [CreateMultipartUpload](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [CompleteMultipartUpload](#)
- [AbortMultipartUpload](#)
- [ListParts](#)
- [ListMultipartUploads](#)

Beispiele

Um einen mehrteiligen Upload zu verwenden, um ein Objekt in die S3 Express One Zone in einem Verzeichnis-Bucket hochzuladen, sehen Sie sich die folgenden Beispiele an.

Themen

- [Erstellen eines mehrteiligen Uploads](#)
- [Teile eines mehrteiligen Uploads hochladen](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Abbrechen eines mehrteiligen Uploads](#)
- [Erstellen eines mehrteiligen Upload-Kopiervorgangs](#)

- [Liste laufender mehrteiliger Uploads](#)
- [Auflisten der Teile eines mehrteiligen Uploads](#)

Erstellen eines mehrteiligen Uploads

Note

Wenn Sie bei Verzeichnis-Buckets einen `CreateMultipartUpload` Vorgang und eine `UploadPartCopy` Operation ausführen, muss die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwenden, und die Anforderungsheader, die Sie in der Anfrage angeben, müssen mit der `CreateMultipartUpload` Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload erstellen.

Verwendung der AWS SDKs

SDK for Java 2.x

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 * that is used to track
 * all the upload parts
 *
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--use1-az4--x-s3'
 * @param key
 * @return
 */
private static String createMultipartUpload(S3Client s3, String bucketName, String
key) {

    CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();
```

```
String uploadId = null;

try {
    CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
    uploadId = response.uploadId();
}
catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return uploadId;
```

SDK for Python

Example

```
def create_multipart_upload(s3_client, bucket_name, key_name):
    """
    Create a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: The destination bucket for the multipart upload
    :param key_name: The key name for the object to be uploaded
    :return: The UploadId for the multipart upload if created successfully, else None
    """

    try:
        mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
key_name)
        return mpu['UploadId']
    except ClientError as e:
        logging.error(e)
        return None
```

Verwendung der AWS CLI

Dieses Beispiel zeigt, wie Sie einen mehrteiligen Upload in einen Verzeichnis-Bucket mithilfe von AWS CLI. Dieser Befehl startet einen mehrteiligen Upload in den Verzeichnis-Bucket *bucket-base-name--azid--x-s3* für das Objekt *KEY_NAME*. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api create-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Weitere Informationen finden Sie [create-multipart-upload](#) in der AWS Command Line Interface.

Teile eines mehrteiligen Uploads hochladen

Die folgenden Beispiele zeigen, wie Teile eines mehrteiligen Uploads hochgeladen werden.

Verwendung der AWS SDKs

SDK for Java 2.x

Das folgende Beispiel zeigt, wie Sie ein einzelnes Objekt in Teile aufteilen und diese Teile dann mithilfe von SDK für Java 2.x in einen Verzeichnis-Bucket hochladen.

Example

```
/**
 * This method creates part requests and uploads individual parts to S3 and then
 * returns all the completed parts
 *
 * @param s3
 * @param bucketName
 * @param key
 * @param uploadId
 * @throws IOException
 */
private static List multipartUpload(S3Client s3, String bucketName,
String key, String uploadId, String filePath) throws IOException {

    int partNumber = 1;
    List completedParts = new ArrayList<>();
    ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

    // read the local file, breakdown into chunks and process
    try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
        long fileSize = file.length();
        int position = 0;
        while (position < fileSize) {
            file.seek(position);
            int read = file.getChannel().read(bb);
```



```

        bb.flip(); // Swap position and limit before reading from the
buffer.
        UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
            .bucket(bucketName)
            .key(key)
            .uploadId(uploadId)
            .partNumber(partNumber)
            .build();

        UploadPartResponse partResponse = s3.uploadPart(
            uploadPartRequest,
            RequestBody.fromByteBuffer(bb));

        CompletedPart part = CompletedPart.builder()
            .partNumber(partNumber)
            .eTag(partResponse.eTag())
            .build();
        completedParts.add(part);

        bb.clear();
        position += read;
        partNumber++;
    }
}

catch (IOException e) {
    throw e;
}
return completedParts;
}

```

SDK for Python

Das folgende Beispiel zeigt, wie Sie ein einzelnes Objekt in Teile aufteilen und diese Teile dann mithilfe von SDK for Python in einen Verzeichnis-Bucket hochladen.

Example

```

def multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_size):
    """
    Break up a file into multiple parts and upload those parts to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    """

```

```

:param key_name: Key name for object to be uploaded and for the local file
that's being uploaded
:param mpu_id: The UploadId returned from the create_multipart_upload call
:param part_size: The size parts that the object will be broken into, in bytes.
                  Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
last part of your multipart upload.
:return: part_list for the multipart upload if all parts are uploaded
successfully, else None
'''

part_list = []
try:
    with open(key_name, 'rb') as file:
        part_counter = 1
        while True:
            file_part = file.read(part_size)
            if not len(file_part):
                break
            upload_part = s3_client.upload_part(
                Bucket = bucket_name,
                Key = key_name,
                UploadId = mpu_id,
                Body = file_part,
                PartNumber = part_counter
            )
            part_list.append({'PartNumber': part_counter, 'ETag':
upload_part['ETag']})
            part_counter += 1
except ClientError as e:
    logging.error(e)
    return None
return part_list

```

Verwendung der AWS CLI

Dieses Beispiel zeigt, wie Sie ein einzelnes Objekt in Teile aufteilen und diese Teile dann mit dem Befehl in einen Verzeichnis-Bucket hochladen AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```

aws s3api upload-part --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --part-number 1 --body LOCAL_FILE_NAME --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA@AAAAAAAAAAAH2AfYAA"

```

Weitere Informationen finden Sie unter [Upload-Teil im AWS Command Line Interface](#).

Abschließen eines mehrteiligen Uploads

Die folgenden Beispiele zeigen, wie ein mehrteiliger Upload abgeschlossen wird.

Verwendung der AWS SDKs

SDK for Java 2.x

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload mithilfe von SDK für Java 2.x abschließen.

Example

```
/**
 * This method completes the multipart upload request by collating all the upload
 parts
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--usw2-az1--x-s3'
 * @param key
 * @param uploadId
 * @param uploadParts
 */
private static void completeMultipartUpload(S3Client s3, String bucketName, String
key, String uploadId, List<CompletedPart> uploadParts) {
    CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
        .parts(uploadParts)
        .build();

    CompleteMultipartUploadRequest completeMultipartUploadRequest =
CompleteMultipartUploadRequest.builder()
        .bucket(bucketName)
        .key(key)
        .uploadId(uploadId)
        .multipartUpload(completedMultipartUpload)
        .build();

    s3.completeMultipartUpload(completeMultipartUploadRequest);
}

public static void multipartUploadTest(S3Client s3, String bucketName, String
key, String localFilePath) {
```

```
System.out.println("Starting multipart upload for: " + key);
try {
    String uploadId = createMultipartUpload(s3, bucketName, key);
    System.out.println(uploadId);
    List<CompletedPart> parts = multipartUpload(s3, bucketName, key, uploadId,
localFilePath);
    completeMultipartUpload(s3, bucketName, key, uploadId, parts);
    System.out.println("Multipart upload completed for: " + key);
}

catch (Exception e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}
```

SDK for Python

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload mithilfe von SDK for Python abschließen.

Example

```
def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    """
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: The destination bucket for the multipart upload
    :param key_name: The key name for the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: The list of uploaded part numbers with their associated ETags
    :return: True if the multipart upload was completed successfully, else False
    """

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
```

```

    )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'OBJECT_NAME'
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_upload(s3_client, bucket_name, key_name, mpu_id,
part_size)
        if part_list is not None:
            if complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id,
part_list):
                print (f'{key_name} successfully uploaded through a ultipart upload
to {bucket_name}')
            else:
                print (f'Could not upload {key_name} hrough a multipart upload to
{bucket_name}')
```

Verwendung der AWS CLI

Dieses Beispiel zeigt, wie Sie einen mehrteiligen Upload für einen Verzeichnis-Bucket mithilfe von AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api complete-multipart-upload --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
--multipart-upload file://parts.json
```

In diesem Beispiel wird eine JSON Struktur verwendet, die die Teile des mehrteiligen Uploads beschreibt, die wieder zur vollständigen Datei zusammengefügt werden sollen. In diesem Beispiel wird das `file://` Präfix verwendet, um die JSON Struktur aus einer Datei im lokalen Ordner mit dem Namen zu laden. `parts`

parts.json:

```
parts.json
{
  "Parts": [
    {
      "ETag": "6b78c4a64dd641a58dac8d9258b88147",
      "PartNumber": 1
    }
  ]
}
```

Weitere Informationen finden Sie [complete-multipart-upload](#) in der AWS Command Line Interface.

Abbrechen eines mehrteiligen Uploads

Die folgenden Beispiele zeigen, wie ein mehrteiliger Upload abgebrochen wird.

Verwendung der AWS SDKs

SDK for Java 2.x

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload mithilfe von für Java 2.x abgebrochen wird. SDK

Example

```
public static void abortMultiPartUploads( S3Client s3, String bucketName ) {

    try {
        ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
            .bucket(bucketName)
            .build();

        ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
        ListMultipartUpload uploads = response.uploads();

        AbortMultipartUploadRequest abortMultipartUploadRequest;
        for (MultipartUpload upload: uploads) {
            abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
                .bucket(bucketName)
```

```
        .key(upload.key())
        .uploadId(upload.uploadId())
        .build();

        s3.abortMultipartUpload(abortMultipartUploadRequest);
    }

}

catch (S3Exception e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}
```

SDK for Python

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload mithilfe von SDK for Python abgebrochen wird.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
    """
    Aborts a partial multipart upload in a directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket where the multipart upload was initiated - for
    example, 'doc-example-bucket--usw2-az1--x-s3'
    :param key_name: Name of the object for which the multipart upload needs to be
    aborted
    :param upload_id: Multipart upload ID for the multipart upload to be aborted
    :return: True if the multipart upload was successfully aborted, False if not
    """
    try:
        s3_client.abort_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = upload_id
```

```

    )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
    upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    if abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
        print (f'Multipart upload for object {key_name} in {bucket_name} bucket has
been aborted')
    else:
        print (f'Unable to abort multipart upload for object {key_name} in
{bucket_name} bucket')

```

Verwendung der AWS CLI

Das folgende Beispiel zeigt, wie Sie einen mehrteiligen Upload mit dem abbrechen AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```

aws s3api abort-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
--upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
MAQAAAAB00xUFeA7LTbWWFS8WYwhrxDxTIDN-pdEEq_agIHqsbg"

```

Weitere Informationen finden Sie [abort-multipart-upload](#) in der AWS Command Line Interface.

Erstellen eines mehrteiligen Upload-Kopiervorgangs

Note

- Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [Von AWS verwalteter Schlüssel](#) (aws/s3) wird nicht unterstützt.

Ihre KMS Konfiguration kann für die gesamte Lebensdauer des Buckets nur einen vom [Kunden verwalteten Schlüssel](#) pro SSE Verzeichnis-Bucket unterstützen. Nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben haben, können Sie den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben. Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den [UploadPartCopy](#) Anforderungsheadern angeben. Außerdem müssen die Anforderungsheader, die Sie in der CreateMultipartUpload Anfrage angeben, der Standardverschlüsselungskonfiguration des Ziel-Buckets entsprechen.

- S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE [UploadPartCopy](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS - verschlüsseltes Objekt gestellt wird.

Die folgenden Beispiele zeigen, wie Objekte mithilfe eines mehrteiligen Uploads von einem Bucket in einen anderen kopiert werden.

Verwendung der AWS SDKs

SDK for Java 2.x

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload verwendet wird, um mithilfe von für Java 2.x ein Objekt programmgesteuert von einem Bucket in einen anderen zu kopieren. SDK

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 * that is used to track
 * all the upload parts.
 *
 * @param s3
 * @param bucketName
 * @param key
 * @return
 */
private static String createMultipartUpload(S3Client s3, String bucketName, String
key) {
```

```

        CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
            .bucket(bucketName)
            .key(key)
            .build();
        String uploadId = null;
        try {
            CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
            uploadId = response.uploadId();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return uploadId;
    }

/**
 * Creates copy parts based on source object size and copies over individual parts
 *
 * @param s3
 * @param sourceBucket
 * @param sourceKey
 * @param destnBucket
 * @param destnKey
 * @param uploadId
 * @return
 * @throws IOException
 */
    public static List<CompletedPart> multipartUploadCopy(S3Client s3, String
sourceBucket, String sourceKey, String destnBucket, String destnKey, String
uploadId) throws IOException {

        // Get the object size to track the end of the copy operation.
        HeadObjectRequest headObjectRequest = HeadObjectRequest
            .builder()
            .bucket(sourceBucket)
            .key(sourceKey)
            .build();
        HeadObjectResponse response = s3.headObject(headObjectRequest);
        Long objectSize = response.contentLength();

        System.out.println("Source Object size: " + objectSize);

```

```
// Copy the object using 20 MB parts.
long partSize = 20 * 1024 * 1024;
long bytePosition = 0;
int partNum = 1;
List<CompletedPart> completedParts = new ArrayList<>();
while (bytePosition < objectSize) {
    // The last part might be smaller than partSize, so check to make sure
    // that lastByte isn't beyond the end of the object.
    long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

    System.out.println("part no: " + partNum + ", bytePosition: " +
bytePosition + ", lastByte: " + lastByte);

    // Copy this part.
    UploadPartCopyRequest req = UploadPartCopyRequest.builder()
        .uploadId(uploadId)
        .sourceBucket(sourceBucket)
        .sourceKey(sourceKey)
        .destinationBucket(destnBucket)
        .destinationKey(destnKey)
        .copySourceRange("bytes="+bytePosition+"-"+lastByte)
        .partNumber(partNum)
        .build();
    UploadPartCopyResponse res = s3.uploadPartCopy(req);
    CompletedPart part = CompletedPart.builder()
        .partNumber(partNum)
        .eTag(res.copyPartResult().eTag())
        .build();
    completedParts.add(part);
    partNum++;
    bytePosition += partSize;
}
return completedParts;
}

public static void multipartCopyUploadTest(S3Client s3, String srcBucket, String
srcKey, String destnBucket, String destnKey) {
    System.out.println("Starting multipart copy for: " + srcKey);
    try {
        String uploadId = createMultipartUpload(s3, destnBucket, destnKey);
        System.out.println(uploadId);
        List<CompletedPart> parts = multipartUploadCopy(s3, srcBucket,
srcKey, destnBucket, destnKey, uploadId);
```

```
        completeMultipartUpload(s3, destnBucket, destnKey, uploadId, parts);
        System.out.println("Multipart copy completed for: " + srcKey);
    } catch (Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
```

SDK for Python

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload verwendet wird, um mithilfe von for Python ein Objekt programmgesteuert von einem Bucket in einen anderen zu kopieren. SDK

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def head_object(s3_client, bucket_name, key_name):
    """
    Returns metadata for an object in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains the object to query for metadata
    :param key_name: Key name to query for metadata
    :return: Metadata for the specified object if successful, else None
    """

    try:
        response = s3_client.head_object(
            Bucket = bucket_name,
            Key = key_name
        )
        return response
    except ClientError as e:
        logging.error(e)
        return None

def create_multipart_upload(s3_client, bucket_name, key_name):
    """
    Create a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    """
```

```
:param bucket_name: Destination bucket for the multipart upload
:param key_name: Key name of the object to be uploaded
:return: UploadId for the multipart upload if created successfully, else None
'''

try:
    mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
key_name)
    return mpu['UploadId']
except ClientError as e:
    logging.error(e)
    return None

def multipart_copy_upload(s3_client, source_bucket_name, key_name,
target_bucket_name, mpu_id, part_size):
    '''
    Copy an object in a directory bucket to another bucket in multiple parts of a
specified size

    :param s3_client: boto3 S3 client
    :param source_bucket_name: Bucket where the source object exists
    :param key_name: Key name of the object to be copied
    :param target_bucket_name: Destination bucket for copied object
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
        Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
last part of your multipart upload.
    :return: part_list for the multipart copy if all parts are copied successfully,
else None
    '''

    part_list = []
    copy_source = {
        'Bucket': source_bucket_name,
        'Key': key_name
    }
    try:
        part_counter = 1
        object_size = head_object(s3_client, source_bucket_name, key_name)
        if object_size is not None:
            object_size = object_size['ContentLength']
            while (part_counter - 1) * part_size < object_size:
                bytes_start = (part_counter - 1) * part_size
                bytes_end = (part_counter * part_size) - 1
```

```
        upload_copy_part = s3_client.upload_part_copy (
            Bucket = target_bucket_name,
            CopySource = copy_source,
            CopySourceRange = f'bytes={bytes_start}-{bytes_end}',
            Key = key_name,
            PartNumber = part_counter,
            UploadId = mpu_id
        )
        part_list.append({'PartNumber': part_counter, 'ETag':
upload_copy_part['CopyPartResult']['ETag']})
        part_counter += 1
    except ClientError as e:
        logging.error(e)
        return None
    return part_list

def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    """
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: List of uploaded part numbers with associated ETags
    :return: True if the multipart upload was completed successfully, else False
    """

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
```

```

region = 'us-west-2'
source_bucket_name = 'SOURCE_BUCKET_NAME'
target_bucket_name = 'TARGET_BUCKET_NAME'
key_name = 'KEY_NAME'
part_size = 10 * MB
s3_client = boto3.client('s3', region_name = region)
mpu_id = create_multipart_upload(s3_client, target_bucket_name, key_name)
if mpu_id is not None:
    part_list = multipart_copy_upload(s3_client, source_bucket_name, key_name,
target_bucket_name, mpu_id, part_size)
    if part_list is not None:
        if complete_multipart_upload(s3_client, target_bucket_name, key_name,
mpu_id, part_list):
            print (f'{key_name} successfully copied through multipart copy from
{source_bucket_name} to {target_bucket_name}')
        else:
            print (f'Could not copy {key_name} through multipart copy from
{source_bucket_name} to {target_bucket_name}')

```

Verwendung der AWS CLI

Das folgende Beispiel zeigt, wie Sie mithilfe eines mehrteiligen Uploads ein Objekt programmgesteuert von einem Bucket in einen Verzeichnis-Bucket kopieren können. Verwenden Sie dazu den AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```

aws s3api upload-part-copy --bucket bucket-base-name--azid--x-s3 --key TARGET_KEY_NAME
--copy-source SOURCE_BUCKET_NAME/SOURCE_KEY_NAME --part-number 1 --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA0AAAAAAAAAAH2AfYAA"

```

Weitere Informationen finden Sie [upload-part-copy](#) in der AWS Command Line Interface.

Liste laufender mehrteiliger Uploads

Um in Bearbeitung befindliche mehrteilige Uploads in einen Verzeichnis-Bucket aufzulisten, können Sie den AWS SDKs, oder AWS CLI.

Verwendung der AWS SDKs

SDK for Java 2.x

Die folgenden Beispiele zeigen, wie in Bearbeitung befindliche (unvollständige) mehrteilige Uploads mithilfe von SDK für Java 2.x aufgelistet werden.

Example

```
public static void listMultiPartUploads( S3Client s3, String bucketName) {
    try {
        ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
        .bucket(bucketName)
        .build();

        ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
        List MultipartUpload uploads = response.uploads();
        for (MultipartUpload upload: uploads) {
            System.out.println("Upload in progress: Key = \"" + upload.key() +
"\", id = " + upload.uploadId());
        }
    }
    catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
```

SDK for Python

Die folgenden Beispiele zeigen, wie in Bearbeitung befindliche (unvollständige) mehrteilige Uploads mithilfe von for Python aufgelistet werden. SDK

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_multipart_uploads(s3_client, bucket_name):
```



```

...
List any incomplete multipart uploads in a directory bucket in e specified gion

:param s3_client: boto3 S3 client
:param bucket_name: Bucket to check for incomplete multipart uploads
:return: List of incomplete multipart uploads if there are any, None if not
...

try:
    response = s3_client.list_multipart_uploads(Bucket = bucket_name)
    if 'Uploads' in response.keys():
        return response['Uploads']
    else:
        return None
except ClientError as e:
    logging.error(e)

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
    multipart_uploads = list_multipart_uploads(s3_client, bucket_name)
    if multipart_uploads is not None:
        print (f'There are {len(multipart_uploads)} ncomplete multipart uploads for
{bucket_name}')
    else:
        print (f'There are no incomplete multipart uploads for {bucket_name}')

```

Verwendung der AWS CLI

Die folgenden Beispiele zeigen, wie Sie in Bearbeitung befindliche (unvollständige) mehrteilige Uploads auflisten können, indem Sie AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api list-multipart-uploads --bucket bucket-base-name--azid--x-s3
```

Weitere Informationen finden Sie [list-multipart-uploads](#) in der AWS Command Line Interface.

Auflisten der Teile eines mehrteiligen Uploads

Die folgenden Beispiele zeigen, wie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket aufgelistet werden.

Verwendung der AWS SDKs

SDK for Java 2.x

Die folgenden Beispiele zeigen, wie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket mithilfe SDK von Java 2.x aufgelistet werden.

```
public static void listMultiPartUploadsParts( S3Client s3, String bucketName, String
objKey, String uploadID) {

    try {
        ListPartsRequest listPartsRequest = ListPartsRequest.builder()
            .bucket(bucketName)
            .uploadId(uploadID)
            .key(objKey)
            .build();

        ListPartsResponse response = s3.listParts(listPartsRequest);
        ListPart parts = response.parts();
        for (Part part: parts) {
            System.out.println("Upload in progress: Part number = \"\" +
part.partNumber() + "\", etag = \"\" + part.eTag());
        }

    }

    catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }

}
```

SDK for Python

Die folgenden Beispiele zeigen, wie Sie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket mithilfe von SDK for Python auflisten können.

```
import logging
import boto3
from botocore.exceptions import ClientError
```

```
def list_parts(s3_client, bucket_name, key_name, upload_id):
    '''
    Lists the parts that have been uploaded for a specific multipart upload to a
    directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that multipart uploads parts have been uploaded to
    :param key_name: Name of the object that has parts uploaded
    :param upload_id: Multipart upload ID that the parts are associated with
    :return: List of parts associated with the specified multipart upload, None if
    there are no parts
    '''
    parts_list = []
    next_part_marker = ''
    continuation_flag = True
    try:
        while continuation_flag:
            if next_part_marker == '':
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id
                )
            else:
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id,
                    NextPartMarker = next_part_marker
                )
            if 'Parts' in response:
                for part in response['Parts']:
                    parts_list.append(part)
                if response['IsTruncated']:
                    next_part_marker = response['NextPartNumberMarker']
            else:
                continuation_flag = False
        else:
            continuation_flag = False
    return parts_list
except ClientError as e:
    logging.error(e)
    return None
```

```
if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
    upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    parts_list = list_parts(s3_client, bucket_name, key_name, upload_id)
    if parts_list is not None:
        print (f'{key_name} has {len(parts_list)} parts uploaded to {bucket_name}')
    else:
        print (f'There are no multipart uploads with that upload ID for
        {bucket_name} bucket')
```

Verwendung der AWS CLI

Die folgenden Beispiele zeigen, wie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket mithilfe von AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api list-parts --bucket bucket-base-name--azid--x-s3 --key KEY_NAME --upload-id
"AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAAAEMAAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAAAA0AAAAAAAAAAAAH2AfYAA
```

Weitere Informationen finden Sie in der [Liste der Teile](#) AWS Command Line Interface.

Hochladen eines Objekts zu einem Verzeichnis-Bucket

Die Kopieroperation erzeugt eine Kopie eines Objekts, das bereits in Amazon S3 gespeichert ist. Sie können Objekte zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Sie können Objekte auch innerhalb eines Buckets und zwischen Buckets desselben Typs kopieren, z. B. von Verzeichnis-Bucket zu Verzeichnis-Bucket.

Sie können eine Kopie eines Objekts mit einer Größe von bis zu 5 GB in einem einzigen atomaren Vorgang erstellen. Um jedoch ein Objekt zu kopieren, das größer als 5 GB ist, müssen Sie die mehrteiligen API Upload-Operationen verwenden. Weitere Informationen finden Sie unter [Mehrteilige Uploads mit Verzeichnis-Buckets verwenden](#).

Berechtigungen

Um Objekte zu kopieren, müssen Sie über die folgenden Berechtigungen verfügen:

- Um Objekte von einem Verzeichnis-Bucket zu einem anderen Verzeichnis-Bucket zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung.
- Um Objekte aus Verzeichnis-Buckets zu Allzweck-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung und `s3:PutObject`-Berechtigung zum Schreiben der Objektkopie in den Ziel-Bucket.
- Um Objekte aus Allzweck-Buckets in Verzeichnis-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession` `s3:GetObject` Erlaubnis und das Recht, das kopierte Quellobjekt zu lesen.

Weitere Informationen finden Sie unter [CopyObject](#) in der Amazon Simple Storage Service API Reference.

Verschlüsselung

Amazon S3 verschlüsselt automatisch alle neuen Objekte, die in einen S3-Bucket hochgeladen werden. Die Standardverschlüsselungskonfiguration eines S3-Buckets ist immer aktiviert und mindestens auf serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) eingestellt.

Für Verzeichnis-Buckets, SSE -S3 und serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) werden unterstützt. Wenn es sich bei dem Ziel-Bucket um einen Verzeichnis-Bucket handelt, empfehlen wir, dass die Standardverschlüsselung des Ziel-Buckets die gewünschte Verschlüsselungskonfiguration verwendet und Sie die Standardverschlüsselung des Buckets nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Außerdem werden S3-Bucket-Keys nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE [CopyObject](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird. Weitere Hinweise zum Verhalten beim Außerkraftsetzen der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung [angeben mit AWS KMS für Uploads neuer Objekte](#).

Für allgemeine Buckets können Sie SSE -S3 (Standard) verwenden, serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE

Wenn Sie eine Kopieranforderung stellen, die angibt, DSSE - KMS oder SSE -C für einen Verzeichnis-Bucket (entweder den Quell- oder den Ziel-Bucket) zu verwenden, gibt die Antwort einen Fehler zurück.

Tags

Verzeichnis-Buckets unterstützen keine Tags. Wenn Sie ein Objekt mit Tags aus einem Allzweck-Bucket in einen Verzeichnis-Bucket kopieren, erhalten Sie eine HTTP 501 (Not Implemented) Antwort. Weitere Informationen finden Sie unter [CopyObject](#) in der Amazon Simple Storage Service API Reference.

ETags

Entitäts-Tags (ETags) für S3 Express One Zone sind zufällige alphanumerische Zeichenfolgen und keine MD5 Prüfsummen. Verwenden Sie zusätzliche Prüfsummen, um die Objektintegrität sicherzustellen.

Zusätzliche Prüfsummen

S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität mit Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) auswählen: CRC32, CRC32C, SHA -1 und -256. SHA MD5 Auf Prüfsummen basierende Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

Weitere Informationen finden Sie unter [Bewährte Methoden für zusätzliche S3-Prüfsummen](#).

Unterstützte Features

Weitere Informationen darüber, welche Amazon S3-Funktionen für S3 Express One Zone unterstützt werden, finden Sie unter [Wodurch zeichnet sich S3 Express One Zone aus?](#).

Verwenden der S3-Konsole (Kopieren in einen Verzeichnis-Bucket)

Um ein Objekt aus einem Allzweck-Bucket oder einem Verzeichnis-Bucket in einen Verzeichnis-Bucket zu kopieren

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.

3. Wählen Sie den Bucket aus, aus dem Sie Objekte kopieren möchten:
 - Um aus einem Allzweck-Bucket zu kopieren, wählen Sie die Registerkarte Allzweck-Buckets.
 - Um aus einem Verzeichnis-Bucket zu kopieren, wählen Sie den Tab Directory-Buckets.
4. Wählen Sie den Allzweck-Bucket oder den Verzeichnis-Bucket, der die Objekte enthält, die Sie kopieren möchten.
5. Wählen Sie die Objekte-Tag. Aktivieren Sie auf der Seite Objekte das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.
6. Wählen Sie im Menü Actions (Aktionen) die Option Copy (Kopieren) aus.

Die Seite Kopieren wird angezeigt.

7. Wählen Sie unter Ziel die Option Directory-Bucket für Ihren Zieltyp aus. Um den Zielpfad anzugeben, wählen Sie Browse S3 aus, navigieren Sie zum Ziel und wählen Sie dann das Optionsfeld links neben dem Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

Geben Sie alternativ den Zielpfad ein.

8. Wählen Sie unter Prüfsummen aus, ob Sie die Objekte mit ihren vorhandenen Prüfsummenfunktionen kopieren oder die vorhandenen Prüfsummenfunktionen durch eine neue ersetzen möchten. Beim Hochladen der Objekte hatten Sie die Möglichkeit, den Prüfsummenalgorithmus anzugeben, der zur Überprüfung der Datenintegrität verwendet wurde. Beim Kopieren des Objekts haben Sie die Möglichkeit, eine neue Funktion auszuwählen. Wenn Sie ursprünglich keine zusätzliche Prüfsumme angegeben haben, können Sie den Abschnitt „Prüfsummen“ verwenden, um eine hinzuzufügen.

Note

Selbst wenn Sie dieselbe Prüfsummenfunktion verwenden, kann sich Ihr Prüfsummenwert ändern, wenn das Objekt größer als 16 MB ist. Der Prüfsummenwert kann sich aufgrund der Methode ändern, wie Prüfsummen für mehrteilige Uploads berechnet werden. Weitere Informationen dazu, wie sich die Prüfsumme beim Kopieren des Objekts ändern kann, finden Sie unter [Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads](#).

Um die Prüfsummenfunktion zu ändern, wählen Sie Replace with a new checksum function (Durch eine neue Prüfsummenfunktion ersetzen) aus. Wählen Sie die neue Prüfsummenfunktion

aus der Drop-down-Liste aus. Wenn das Objekt kopiert wird, wird die neue Prüfsumme unter Verwendung des angegebenen Algorithmus berechnet und gespeichert.

9. Wählen Sie unten rechts Copy (Kopieren) aus. Amazon S3 kopiert Ihre Objekte in den Zielordner.

Verwenden der S3-Konsole (Kopieren zu einem Allzweck-Bucket)

So kopieren Sie ein Objekt aus einem Verzeichnis-Bucket in einen Allzweck-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Directory-Buckets.
4. Wählen Sie den Verzeichnis-Bucket aus, der die Objekte enthält, die Sie kopieren möchten.
5. Wählen Sie die Objekte-Tag. Aktivieren Sie auf der Seite Objekte das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.
6. Wählen Sie im Menü Actions (Aktionen) die Option Copy (Kopieren) aus.
7. Wählen Sie unter Ziel die Option Allzweck-Bucket für Ihren Zieltyp aus. Um den Zielpfad anzugeben, wählen Sie Browse S3 aus, navigieren Sie zum Ziel und wählen Sie das Optionsfeld links neben dem Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

Geben Sie alternativ den Zielpfad ein.

8. Wählen Sie unter Prüfsummen aus, ob Sie die Objekte mit ihren vorhandenen Prüfsummenfunktionen kopieren oder die vorhandenen Prüfsummenfunktionen durch eine neue ersetzen möchten. Beim Hochladen der Objekte hatten Sie die Möglichkeit, den Prüfsummenalgorithmus anzugeben, der zur Überprüfung der Datenintegrität verwendet wurde. Beim Kopieren des Objekts haben Sie die Möglichkeit, eine neue Funktion auszuwählen. Wenn Sie ursprünglich keine zusätzliche Prüfsumme angegeben haben, können Sie den Abschnitt Prüfsummen verwenden, um eine hinzuzufügen.

Note

Selbst wenn Sie dieselbe Prüfsummenfunktion verwenden, kann sich Ihr Prüfsummenwert ändern, wenn das Objekt größer als 16 MB ist. Der Prüfsummenwert kann sich aufgrund der Methode ändern, wie Prüfsummen für mehrteilige Uploads

berechnet werden. Weitere Informationen dazu, wie sich die Prüfsumme beim Kopieren des Objekts ändern kann, finden Sie unter [Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads](#).

Um die Prüfsummenfunktion zu ändern, wählen Sie Replace with a new checksum function (Durch eine neue Prüfsummenfunktion ersetzen) aus. Wählen Sie die neue Prüfsummenfunktion aus der Drop-down-Liste aus. Wenn das Objekt kopiert wird, wird die neue Prüfsumme unter Verwendung des angegebenen Algorithmus berechnet und gespeichert.

9. Wählen Sie unten rechts Copy (Kopieren) aus. Amazon S3 kopiert Ihre Objekte in den Zielordner.

Verwendung der AWS SDKs

SDK for Java 2.x

Example

```
public static void copyBucketObject (S3Client s3, String sourceBucket, String
objectKey, String targetBucket) {
    CopyObjectRequest copyReq = CopyObjectRequest.builder()
        .sourceBucket(sourceBucket)
        .sourceKey(objectKey)
        .destinationBucket(targetBucket)
        .destinationKey(objectKey)
        .build();
    String temp = "";

    try {
        CopyObjectResponse copyRes = s3.copyObject(copyReq);
        System.out.println("Successfully copied " + objectKey + " from bucket " +
sourceBucket + " into bucket "+targetBucket);
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Verwendung der AWS CLI

Der folgende `copy-object` Beispielbefehl zeigt, wie Sie AWS CLI um ein Objekt von einem Bucket in einen anderen Bucket zu kopieren. Sie können Objekte zwischen Bucket-Typen kopieren. Um diesen Befehl auszuführen, ersetzen Sie die Platzhalter für Benutzereingaben durch Ihre eigenen Informationen.

```
aws s3api copy-object --copy-source bucket SOURCE_BUCKET/SOURCE_KEY_NAME --  
key TARGET_KEY_NAME --bucket TARGET_BUCKET_NAME
```

Weitere Informationen finden Sie unter [copy-object](#) in der AWS CLI Befehlsreferenz.

Löschen eines Objekts in einem Verzeichnis-Bucket

Sie können Objekte aus einem Amazon S3-Verzeichnis-Bucket löschen, indem Sie die Amazon S3 S3-Konsole AWS Command Line Interface (AWS CLI) oder AWS SDKs verwenden. Weitere Informationen finden Sie unter [Verzeichnis-Buckets](#) und [Was ist S3 Express One Zone?](#).

Warning

- Das Löschen eines Objekts kann nicht rückgängig gemacht werden.
- Diese Aktion löscht alle angegebenen Objekte. Warten Sie beim Löschen von Ordnern, bis die Löschaktion abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte gelöscht werden.

Note

Beachten Sie Folgendes, wenn Sie programmgesteuert mehrere Objekte aus einem Verzeichnis-Bucket löschen:

- Objektschlüssel in `DeleteObjects`-Anforderungen müssen mindestens ein Zeichen enthalten, das kein Leerzeichen ist. Zeichenketten, die ausschließlich Leerzeichen enthalten, werden nicht unterstützt.
- Objektschlüssel in `DeleteObjects` Anfragen dürfen keine Unicode-Steuerzeichen enthalten, mit Ausnahme von Newline (`\n`), Tab (`\t`) und Carriage Return (`\r`).

Verwenden der S3-Konsole

So löschen Sie Objekte

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie den Verzeichnis-Bucket aus, der die Objekte enthält, die Sie löschen möchten.
5. Wählen Sie die Objekte-Tag. Aktivieren Sie in der Objektliste das Kontrollkästchen links neben dem Objekt oder den Objekten, die Sie löschen möchten.
6. Wählen Sie Löschen aus.
7. Geben Sie auf der Seite Objekte löschen den Text **permanently delete** in das Textfeld ein.
8. Wählen Sie Delete objects (Objekte löschen).

Verwenden der AWS SDKs

SDK for Java 2.x

Example

Im folgenden Beispiel werden Objekte in einem Verzeichnis-Bucket mithilfe von gelöscht. AWS SDK for Java 2.x

```
static void deleteObject(S3Client s3Client, String bucketName, String objectKey) {  
  
    try {  
  
        DeleteObjectRequest del = DeleteObjectRequest.builder()  
            .bucket(bucketName)  
            .key(objectKey)  
            .build();  
  
        s3Client.deleteObject(del);  
  
        System.out.println("Object " + objectKey + " has been deleted");  
    }  
}
```

```
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

Im folgenden Beispiel werden Objekte in einem Verzeichnis-Bucket mithilfe von gelöscht. AWS SDK for Python (Boto3)

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_objects(s3_client, bucket_name, objects):
    """
    Delete a list of objects in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains objects to be deleted; for example,
    'doc-example-bucket--usw2-az1--x-s3'
    :param objects: List of dictionaries that specify the key names to delete
    :return: Response output, else False
    """

    try:
        response = s3_client.delete_objects(
            Bucket = bucket_name,
            Delete = {
                'Objects': objects
            }
        )
        return response
    except ClientError as e:
        logging.error(e)
        return False
```

```
if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    objects = [
        {
            'Key': '0.txt'
        },
        {
            'Key': '1.txt'
        },
        {
            'Key': '2.txt'
        },
        {
            'Key': '3.txt'
        },
        {
            'Key': '4.txt'
        }
    ]

    s3_client = boto3.client('s3', region_name = region)
    results = delete_objects(s3_client, bucket_name, objects)
    if results is not None:
        if 'Deleted' in results:
            print (f'Deleted {len(results["Deleted"])} objects from {bucket_name}')
        if 'Errors' in results:
            print (f'Failed to delete {len(results["Errors"])} objects from
{bucket_name}')
```

Mit dem AWS CLI

Der folgende `delete-object` Beispielbefehl zeigt, wie Sie mit dem AWS CLI ein Objekt aus einem Verzeichnis-Bucket löschen können. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api delete-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Weitere Informationen finden Sie unter [delete-object](#) in der Referenz zum AWS CLI -Befehl.

Ein Objekt in einem Verzeichnis-Bucket herunterladen

Die folgenden Codebeispiele zeigen, wie mithilfe der `GetObject` API-Operation Daten aus einem Objekt in einem Amazon S3 S3-Verzeichnis-Bucket gelesen (heruntergeladen) werden.

Verwendung der AWS SDKs

SDK for Java 2.x

Example

Das folgende Codebeispiel zeigt, wie Daten aus einem Objekt in einem Verzeichnis-Bucket mithilfe von `GetObject` gelesen werden. `AWS SDK for Java 2.x`

```
public static void getObject(S3Client s3Client, String bucketName, String objectKey)
{
    try {
        GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(objectKey)
            .bucket(bucketName)
            .build();

        ResponseBytes GetObjectResponse objectBytes =
s3Client.getObjectAsBytes(objectRequest);
        byte[] data = objectBytes.asByteArray();

        //Print object contents to console
        String s = new String(data, StandardCharsets.UTF_8);
        System.out.println(s);
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

Das folgende Codebeispiel zeigt, wie Daten aus einem Objekt in einem Verzeichnis-Bucket mithilfe von gelesen werden AWS SDK for Python (Boto3).

```
import boto3
from botocore.exceptions import ClientError
from botocore.response import StreamingBody

def get_object(s3_client: boto3.client, bucket_name: str, key_name: str) ->
StreamingBody:
    """
    Gets the object.
    :param s3_client:
    :param bucket_name: The bucket that contains the object.
    :param key_name: The key of the object to be downloaded.
    :return: The object data in bytes.
    """
    try:
        response = s3_client.get_object(Bucket=bucket_name, Key=key_name)
        body = response['Body'].read()
        print(f"Got object '{key_name}' from bucket '{bucket_name}'.")
    except ClientError:
        print(f"Couldn't get object '{key_name}' from bucket '{bucket_name}'.")
        raise
    else:
        return body

def main():
    s3_client = boto3.client('s3')
    resp = get_object(s3_client, 'doc-example-bucket--use1-az4--x-s3', 'sample.txt')
    print(resp)

if __name__ == "__main__":
    main()
```

Mit dem AWS CLI

Das folgende Beispiel für `get-object` veranschaulicht, wie Sie die AWS CLI verwenden können, um ein Objekt von Amazon S3 herunterzuladen. Mit diesem Befehl wird das Objekt **KEY_NAME** aus

dem Verzeichnis-Bucket abgerufen *bucket-base-name--azid--x-s3*. Das Objekt wird in eine Datei mit dem Namen *LOCAL_FILE_NAME* heruntergeladen. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api get-object --bucket bucket-base-name--azid--x-s3 --  
key KEY_NAME LOCAL_FILE_NAME
```

Weitere Informationen finden Sie unter [get-object](#) in der Referenz zum AWS CLI -Befehl.

Verwendung **HeadObject** mit Verzeichnis-Buckets

Die folgenden AWS SDK- und AWS CLI-Beispiele zeigen, wie Sie mithilfe der `HeadObject` API-Operation Metadaten von einem Objekt in einem Amazon S3 S3-Verzeichnis-Bucket abrufen können, ohne das Objekt selbst zurückzugeben.

Verwendung der AWS SDKs

SDK for Java 2.x

Example

```
public static void headObject(S3Client s3Client, String bucketName, String  
objectKey) {  
    try {  
        HeadObjectRequest headObjectRequest = HeadObjectRequest  
            .builder()  
            .bucket(bucketName)  
            .key(objectKey)  
            .build();  
        HeadObjectResponse response = s3Client.headObject(headObjectRequest);  
        System.out.format("Amazon S3 object: \"%s\" found in bucket: \"%s\" with  
ETag: \"%s\"", objectKey, bucketName, response.eTag());  
    }  
    catch (S3Exception e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
    }  
}
```


Mit dem AWS CLI

Der folgende `head-object` Beispielbefehl zeigt, wie Sie mit AWS CLI dem Metadaten aus einem Objekt abrufen können. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api head-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Weitere Informationen finden Sie unter [head-object](#) in der Referenz zum AWS CLI -Befehl.

Sicherheit für Verzeichnis-Buckets

Cloud-Sicherheit bei AWS hat höchste Priorität. Als AWS Als Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen. Sicherheit ist eine gemeinsame Verantwortung zwischen AWS und du. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- **Sicherheit der Cloud** — AWS ist verantwortlich für den Schutz der Infrastruktur, die läuft AWS-Services in der AWS Cloud. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheit im Rahmen der [AWS Compliance Programs](#).

Weitere Informationen zu den Compliance-Programmen, die für Amazon S3 Express One Zone gelten, finden Sie unter [AWS-Services in Scope by Compliance Program](#).

- **Sicherheit in der Cloud** — Ihre Verantwortung richtet sich nach AWS-Service die Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, die Anforderungen Ihres Unternehmens und die geltenden Gesetze und Vorschriften.

Diese Dokumentation beschreibt, wie Sie das Modell der gemeinsamen Verantwortlichkeit bei der Verwendung von S3 Express Zone anwenden können. Die folgenden Themen veranschaulichen, wie Sie S3 Express One Zone zur Erfüllung Ihrer Sicherheits- und Compliance-Anforderungen konfigurieren können. Sie werden auch lernen, wie man andere benutzt AWS-Services das kann Ihnen helfen, Ihre Ressourcen zu überwachen und zu sichern, wenn Sie mit S3 Express One Zone arbeiten.

Themen

- [Datenschutz und Verschlüsselung in S3 Express One Zone](#)

- [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#)
- [Auf IAM-Identitäten basierende IAM-Richtlinien für S3 Express One Zone](#)
- [Beispiel für Verzeichnis-Bucket-Richtlinien für S3 Express One Zone](#)
- [CreateSession-Autorisierung](#)
- [Bewährte Methoden für die Sicherheit in S3 Express One Zone](#)

Datenschutz und Verschlüsselung in S3 Express One Zone

Weitere Informationen darüber, wie S3 Express One Zone Ihre Daten verschlüsselt und schützt, finden Sie in den folgenden Themen.

Themen

- [Server-side encryption](#)
- [Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets](#)
- [Serverseitige Verschlüsselung verwenden mit AWS KMS Schlüssel \(SSE-KMS\) in Verzeichnis-Buckets](#)
- [Verschlüsselung während der Übertragung](#)
- [Zusätzliche Prüfsummen](#)
- [Löschen von Daten](#)

Server-side encryption

Für alle Verzeichnis-Buckets ist standardmäßig die Verschlüsselung konfiguriert, und alle neuen Objekte, die in Directory-Buckets hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Verzeichnis-Bucket. Wenn Sie einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), indem Sie die Standardverschlüsselungskonfiguration des Buckets festlegen. Weitere Hinweise zu SSE - KMS in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung verwenden mit AWS KMS Schlüssel \(SSE-KMS\) in Verzeichnis-Buckets](#).

Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren CreateSession Anfragen oder PUT Objektanforderungen nicht überschreiben. Anschließend

werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).

SSE- KMS mit Verzeichnis-Buckets unterscheidet sich von SSE - KMS in Allzweck-Buckets in den folgenden Aspekten.

- Ihre KMS Konfiguration SSE — unterstützt für die gesamte Lebensdauer des Buckets nur einen vom [Kunden verwalteten Schlüssel](#) pro Verzeichnis-Bucket. [Von AWS verwalteter Schlüssel](#)(aws/s3) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben haben KMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

- Sie stellen eine HeadObject API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

- Bei [API Vorgängen auf zentraler Endpunktebene \(Objektebene\)](#) mit Ausnahme von [CopyObject](#) und [UploadPartCopy](#) authentifizieren und autorisieren Sie Anfragen mit geringer Latenz. [CreateSession](#) Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder Objektanforderungen nicht überschreiben. PUT Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln verschlüsselt und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).

Bei API Aufrufen von zonalen Endpunkten (außer [CopyObject](#) und [UploadPartCopy](#)) können Sie die Werte der Verschlüsselungseinstellungen (`x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, `x-amz-server-side-encryption-context`, und `undx-amz-server-side-encryption-bucket-key-enabled`) aus der `CreateSession` Anforderung nicht überschreiben. Sie müssen diese Werte für die Verschlüsselungseinstellungen nicht explizit in API Aufrufen von zonalen Endpunkten angeben, und Amazon S3 verwendet die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage, um neue Objekte im Verzeichnis-Bucket zu schützen.

Note

Wenn Sie das verwenden AWS CLI oder das AWS SDKs, für `CreateSession`, das Sitzungstoken wird automatisch aktualisiert, um Dienstunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Das Tool AWS CLI oder der AWS SDKs verwendet die Standard-Verschlüsselungskonfiguration des Buckets für die `CreateSession` Anfrage. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen in der `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer [CopyObject](#) und [UploadPartCopy](#)) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben.

- Um neue Objektkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [CopyObject](#) Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objektkopien mit SSE - angeben, müssen Sie sicherstellen KMS, dass der Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Directory-Buckets angegeben haben. Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [UploadPartCopy](#) Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den [UploadPartCopy](#) Anforderungsheadern angeben. Außerdem müssen die Verschlüsselungseinstellungen, die Sie in der [CreateMultipartUpload](#) Anfrage angeben, mit der Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.
- S3-Bucket-Keys sind immer für GET und für PUT Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS

verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE [CopyObject](#), [UploadPartCopy](#), der [Copy Operation in Batch-Operationen](#) oder [die import Jobs](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird.

- Wenn Sie eine angeben [AWS KMS Verwenden Sie](#) für die Verschlüsselung in Ihrem Verzeichnis-Bucket vom Kunden verwalteten Schlüssel nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.

Directory-Buckets unterstützen keine serverseitige Dual-Layer-Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (-C). SSE

Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets

In Amazon S3 S3-Buckets ist die Bucket-Verschlüsselung standardmäßig aktiviert, und neue Objekte werden automatisch mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Diese Verschlüsselung gilt für alle neuen Objekte in Ihren Amazon-S3-Buckets und es fallen keine Kosten für Sie an.

Wenn Sie mehr Kontrolle über Ihre Verschlüsselungsschlüssel benötigen, z. B. die Verwaltung der Schlüsselrotation und die Gewährung von Zugriffsrichtlinien, können Sie sich für die serverseitige Verschlüsselung mit entscheiden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS).

Note

- Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanfragen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).
- Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue

Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln von KMS - und entschlüsselt.

- Wenn Sie die Standard-Bucket-Verschlüsselung auf SSE - setzen KMS, sind S3-Bucket-Keys immer für GET PUT Vorgänge in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE, bis [CopyObject](#), [UploadPartCopy](#), der [Copy Operation in Batch-Operationen](#), oder [die import Jobs](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird. Weitere Informationen darüber, wie S3 Bucket Keys Ihre reduzierten AWS KMS Kosten für Anfragen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).
- Wenn Sie eine angeben [AWS KMS Verwenden Sie](#) für die Verschlüsselung in Ihrem Verzeichnis-Bucket vom Kunden verwalteten Schlüssel nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.
- Zweiseitige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) und serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) werden für die Standardverschlüsselung in Verzeichnis-Buckets nicht unterstützt.

Weitere Hinweise zur Konfiguration der Standardverschlüsselung finden Sie unter [Konfigurieren der Standardverschlüsselung](#)

Weitere Informationen zu den für die Standardverschlüsselung erforderlichen Berechtigungen finden Sie unter [PutBucketEncryption](#) in der Amazon Simple Storage Service API Reference.

Sie können die Amazon S3 S3-Standardverschlüsselung für einen S3-Bucket mithilfe der Amazon S3 S3-Konsole konfigurieren, AWS SDKs, der Amazon S3 REST API und der AWS Command Line Interface (AWS CLI).

Verwenden der S3-Konsole

Konfigurieren der Standardverschlüsselung für einen Amazon-S3-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.

4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Unter Serverseitige Verschlüsselungseinstellungen verwenden Verzeichnis-Buckets serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3).
6. Wählen Sie Änderungen speichern.

Verwendung der AWS CLI

Diese Beispiele zeigen Ihnen, wie Sie die Standardverschlüsselung mithilfe von SSE -S3 oder mithilfe von SSE - KMS mit einem S3-Bucket-Key konfigurieren.

Weitere Informationen zur Standardverschlüsselung finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#). Weitere Informationen zur Verwendung von AWS CLI Informationen zur Konfiguration der Standardverschlüsselung finden Sie unter [put-bucket-encryption](#).

Example — Standardverschlüsselung mit SSE -S3

In diesem Beispiel wird die Standardverschlüsselung von Buckets mit von Amazon S3 verwalteten Schlüsseln konfiguriert. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api put-bucket-encryption --bucket bucket-base-name--azid--x-s3 --server-side-encryption-configuration '{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "AES256"
      }
    }
  ]
}'
```

Example — Standardverschlüsselung mit SSE — KMS unter Verwendung eines S3-Bucket-Keys

In diesem Beispiel wird die Standard-Bucket-Verschlüsselung mit SSE - KMS unter Verwendung eines S3-Bucket-Keys konfiguriert. Um den Befehl zu verwenden, ersetzen Sie *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api put-bucket-encryption --bucket bucket-base-name--azid--x-s3 --server-side-encryption-configuration '{
```

```
"Rules": [
  {
    "ApplyServerSideEncryptionByDefault": {
      "SSEAlgorithm": "aws:kms",
      "KMSEncryptionContext": "KMS-Key-ARN"
    },
    "BucketKeyEnabled": true
  }
]
```

Mit dem REST API

Verwenden Sie den REST API `PutBucketEncryption` Vorgang, um die Standardverschlüsselung mit einem serverseitigen Verschlüsselungstyp festzulegen — SSE -S3 oder SSE - . KMS

Weitere Informationen finden Sie unter [PutBucketEncryption](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Bei der Verwendung AWS SDKs, Sie können Amazon S3 zur Nutzung anfordern AWS KMS keys für serverseitige Verschlüsselung. Folgendes AWS SDKs für Java und .NET Beispiele konfigurieren die Standardverschlüsselungskonfiguration für einen Verzeichnis-Bucket mit SSE - KMS und einem S3-Bucket-Key. Informationen zu anderen SDKs finden Sie unter [Beispielcode und Bibliotheken](#) auf der AWS Entwicklerzentrum.

Important

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Java

Mit dem AWS SDK for Java 2.x, können Sie Amazon S3 auffordern, ein AWS KMS key indem Sie die `applyServerSideEncryptionByDefault` Methode verwenden, um die

Standardverschlüsselungskonfiguration Ihres Verzeichnis-Buckets für die Datenverschlüsselung mit SSE - anzugebenKMS. Sie erstellen einen symmetrischen KMS Verschlüsselungsschlüssel und geben diesen in der Anfrage an.

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutBucketEncryptionRequest;
import software.amazon.awssdk.services.s3.model.ServerSideEncryption;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionByDefault;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionConfiguration;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionRule;

public class Main {
    public static void main(String[] args) {
        S3Client s3 = S3Client.create();
        String bucketName = "bucket-base-name--azid--x-s3";
        String kmsKeyId = "your-kms-customer-managed-key-id";

        // AWS managed KMS keys aren't supported. Only customer-managed keys are
        // supported.
        ServerSideEncryptionByDefault serverSideEncryptionByDefault =
        ServerSideEncryptionByDefault.builder()
            .sseAlgorithm(ServerSideEncryption.AWS_KMS)
            .kmsMasterKeyId(kmsKeyId)
            .build();

        // The bucketKeyEnabled field is enforced to be true.
        ServerSideEncryptionRule rule = ServerSideEncryptionRule.builder()
            .bucketKeyEnabled(true)
            .applyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
            .build();

        ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
        ServerSideEncryptionConfiguration.builder()
            .rules(rule)
            .build();

        PutBucketEncryptionRequest putRequest = PutBucketEncryptionRequest.builder()
            .bucket(bucketName)

            .serverSideEncryptionConfiguration(serverSideEncryptionConfiguration)
            .build();
    }
}
```

```
s3.putBucketEncryption(putRequest);  
  
    }  
}
```

Weitere Informationen zur Erstellung von vom Kunden verwalteten Schlüsseln finden Sie unter [Programmieren der AWS KMS API](#) in der AWS Key Management Service Leitfaden für Entwickler.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter [Hochladen eines Objekts zu einem Verzeichnis-Bucket](#).
- Informationen zu mehrteiligen API Upload-Vorgängen finden Sie unter [Mehrteilige Uploads mit Verzeichnis-Buckets verwenden](#).

.NET

Mit dem AWS SDK for .NET, können Sie Amazon S3 auffordern, ein AWS KMS key indem Sie die `ServerSideEncryptionByDefault` Eigenschaft verwenden, um die Standard-Verschlüsselungskonfiguration Ihres Verzeichnis-Buckets für die Datenverschlüsselung mit SSE - anzugebenKMS. Sie erstellen einen vom Kunden verwalteten symmetrischen Verschlüsselungsschlüssel und geben diesen in der Anfrage an.

```
// Set the bucket server side encryption to use AWSKMS with a customer-managed  
key id.  
// bucketName: Name of the directory bucket. "bucket-base-name--azid--x-s3"  
// kmsKeyId: The Id of the customer managed KMS Key. "your-kms-customer-managed-  
key-id"  
// Returns True if successful.  
public static async Task<bool> SetBucketServerSideEncryption(string bucketName,  
string kmsKeyId)  
{  
    var serverSideEncryptionByDefault = new ServerSideEncryptionConfiguration  
{  
        ServerSideEncryptionRules = new List<ServerSideEncryptionRule>  
{
```

```
        new ServerSideEncryptionRule
        {
            ServerSideEncryptionByDefault = new
ServerSideEncryptionByDefault
            {
                ServerSideEncryptionAlgorithm =
ServerSideEncryptionMethod.AWSKMS,
                ServerSideEncryptionKeyManagementServiceKeyId = kmsKeyId
            }
        }
    };
    try
    {
        var encryptionResponse =await _s3Client.PutBucketEncryptionAsync(new
PutBucketEncryptionRequest
        {
            BucketName = bucketName,
            ServerSideEncryptionConfiguration = serverSideEncryptionByDefault,
        });

        return encryptionResponse.HttpStatusCode == HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine(ex.ErrorCode == "AccessDenied"
            ? $"This account does not have permission to set encryption on
{bucketName}, please try again."
            : $"Unable to set bucket encryption for bucket {bucketName},
{ex.Message}");
    }
    return false;
}
```

Weitere Informationen zur Erstellung von vom Kunden verwalteten Schlüsseln finden Sie unter [Programmieren der AWS KMS API](#) in der AWS Key Management Service Leitfaden für Entwickler.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter [Hochladen eines Objekts zu einem Verzeichnis-Bucket](#).
- Informationen zu mehrteiligen API Upload-Vorgängen finden Sie unter [Mehrteilige Uploads mit Verzeichnis-Buckets verwenden](#).

Überwachung der Standardverschlüsselung für Verzeichnis-Buckets mit AWS CloudTrail

Sie können Anfragen zur Standardverschlüsselungskonfiguration für Amazon S3-Verzeichnis-Buckets verfolgen, indem Sie AWS CloudTrail Ereignisse. Die folgenden API Ereignisnamen werden in CloudTrail Protokollen verwendet:

- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

Note

- EventBridge wird in Verzeichnis-Buckets nicht unterstützt.
- Zweiseichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) werden in Verzeichnis-Buckets nicht unterstützt.

Weitere Informationen zur Überwachung der Standardverschlüsselung mit AWS CloudTrail, finden Sie unter [Überwachung der Standardverschlüsselung mit AWS CloudTrail und Amazon EventBridge](#).

Serverseitige Verschlüsselung verwenden mit AWS KMS Schlüssel (SSE-KMS) in Verzeichnis-Buckets

Die Sicherheitskontrollen in AWS KMS kann Ihnen helfen, die Compliance-Anforderungen im Zusammenhang mit der Verschlüsselung zu erfüllen. Sie können sich dafür entscheiden, Verzeichnis-Buckets für die serverseitige Verschlüsselung zu konfigurieren AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) und verwenden Sie diese KMS Schlüssel, um Ihre Daten in Amazon S3-Verzeichnis-Buckets zu schützen. Weitere Informationen zu SSE - finden Sie KMS unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

Berechtigungen

Um ein Objekt hoch- oder herunterzuladen, das mit einem verschlüsselt ist AWS KMS key zu oder von Amazon S3 benötigen `kms:GenerateDataKey` Sie `kms:Decrypt` Berechtigungen für den Schlüssel. Weitere Informationen finden Sie unter [Erlauben Sie Schlüsselbenutzern, einen KMS Schlüssel für kryptografische Operationen zu verwenden](#) in der AWS Key Management Service Leitfaden für Entwickler. Für Informationen über AWS KMS Berechtigungen, die für mehrteilige Uploads erforderlich sind, finden Sie unter [Mehrteiliger Upload API und Berechtigungen](#)

Weitere Hinweise zu KMS Schlüsseln für SSE - finden Sie KMS unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#)

Themen

- [AWS KMS keys](#)
- [Verwenden von SSE - KMS für kontenübergreifende Operationen](#)
- [Amazon-S3-Bucket-Schlüssel](#)
- [Erfordert SSE - KMS](#)
- [Verschlüsselungskontext](#)
- [Senden von Anfragen für AWS KMS verschlüsselte Objekte](#)
- [Auditing SSE — KMS Verschlüsselung in Verzeichnis-Buckets](#)
- [Serverseitige Verschlüsselung angeben mit AWS KMS \(SSE-KMS\) für das Hochladen neuer Objekte in Verzeichnis-Buckets](#)

AWS KMS keys

Ihre KMS Konfiguration SSE — unterstützt für die gesamte Lebensdauer des Buckets nur einen vom [Kunden verwalteten Schlüssel](#) pro Verzeichnis-Bucket. [Von AWS verwalteter Schlüssel](#)(aws/s3) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben haben KMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

- Sie stellen eine `HeadObject` API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

Wenn Sie eine angeben [AWS KMS Verwenden Sie](#) für die Verschlüsselung in Ihrem Verzeichnis-Bucket vom Kunden verwalteten Schlüssel nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.

Weitere Hinweise zu KMS Schlüsseln für SSE - KMS finden Sie unter [AWS KMS keys](#).

Verwenden von SSE - KMS für kontenübergreifende Operationen

Wenn Sie Verschlüsselung für kontoübergreifende Operationen in Verzeichnis-Buckets verwenden, sollten Sie Folgendes beachten:

- Wenn Sie kontoübergreifenden Zugriff auf Ihre S3-Objekte gewähren möchten, konfigurieren Sie eine Richtlinie mit einem vom Kunden verwalteten Schlüssel, um den Zugriff von einem anderen Konto aus zu ermöglichen.
- Um einen vom Kunden verwalteten Schlüssel anzugeben, müssen Sie einen vollqualifizierten KMS Schlüssel ARN verwenden.

Amazon-S3-Bucket-Schlüssel

S3-Bucket-Keys sind immer für GET PUT Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE [CopyObject](#), [UploadPartCopy](#), der [Copy Operation in Batch-Operationen](#) oder [die import Jobs](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird.

Bei [APIVorgängen an zonalen Endpunkten \(Objektebene\)](#) mit Ausnahme von und authentifizieren [CopyObject](#) und [UploadPartCopy](#) autorisieren Sie Anfragen über [CreateSession](#) für niedrige Latenz. Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren [CreateSession](#) Anfragen oder PUT Objektanfragen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit

einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).

S3 Bucket Keys werden für einen begrenzten Zeitraum in Amazon S3 verwendet, wodurch Amazon S3 weiter weniger Anfragen stellen muss AWS KMS um Verschlüsselungsvorgänge abzuschließen. Weitere Informationen zur Verwendung von S3-Bucket Keys finden Sie unter [Amazon-S3-Bucket-Schlüssel](#) und [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Erfordert SSE - KMS

Um SSE - KMS von allen Objekten in einem bestimmten Verzeichnis-Bucket zu verlangen, können Sie eine Bucket-Richtlinie verwenden. Wenn Sie den `CreateSession` API Vorgang beispielsweise verwenden, um die Erlaubnis zum Hochladen eines neuen Objekts (`PutObject`, und `CreateMultipartUpload`) zu erteilen `CopyObject`, verweigert die folgende Bucket-Richtlinie allen Benutzern die Berechtigung zum Hochladen von Objekten (`s3express:CreateSession`), wenn die `CreateSession` Anfrage keinen `x-amz-server-side-encryption-aws-kms-key-id` Header enthält, der SSE - KMS anfordert.

```
{
  "Version": "2012-10-17",
  "Id": "UploadObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSEKMS",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3express:CreateSession",
      "Resource": "arn:aws:s3express:region:account-id:bucket/bucket-base-name--azid--x-s3/*",
      "Condition": {
        "Null": {
          "s3express:x-amz-server-side-encryption-aws-kms-key-id": "true"
        }
      }
    }
  ]
}
```

Um zu verlangen, dass ein bestimmtes AWS KMS key Um die Objekte in einem Bucket zu verschlüsseln, können Sie den `s3express:x-amz-server-side-encryption-aws-kms-key-id` Bedingungsschlüssel verwenden. Um den KMS Schlüssel anzugeben, müssen Sie einen Schlüssel Amazon Resource Name (ARN) verwenden, der das `arn:aws:kms:region:acct-id:key/key-id` Format hat. AWS Identity and Access Management überprüft nicht, ob die Zeichenfolge für `s3express:x-amz-server-side-encryption-aws-kms-key-id` existiert. Das Tool AWS KMS Die Schlüssel-ID, die Amazon S3 für die Objektverschlüsselung verwendet, muss mit der übereinstimmen AWS KMS Schlüssel-ID in der Richtlinie, andernfalls lehnt Amazon S3 die Anfrage ab.

Weitere Informationen zur Verwendung SSE — KMS für Uploads neuer Objekte finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS \(SSE-KMS\) für das Hochladen neuer Objekte in Verzeichnis-Buckets](#)

Eine vollständige Liste der spezifischen Bedingungsschlüssel für Directory-Buckets finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#)

Verschlüsselungskontext

Bei Verzeichnis-Buckets besteht ein Verschlüsselungskontext aus einer Reihe von Schlüssel-Wert-Paaren, die kontextbezogene Informationen zu den Daten enthalten. Ein zusätzlicher Wert für den Verschlüsselungskontext wird nicht unterstützt. Weitere Informationen zum Thema Verschlüsselungskontext finden Sie unter [Verschlüsselungskontext](#).

Wenn Sie SSE - KMS für einen Verzeichnis-Bucket verwenden, verwendet Amazon S3 standardmäßig den Bucket Amazon Resource Name (ARN) als Verschlüsselungskontext-Paar:

```
arn:aws:s3express:region:account-id:bucket/bucket-base-name--azid--x-s3
```

Stellen Sie sicher, dass Ihre IAM Richtlinien oder AWS KMS wichtige Richtlinien verwenden Ihren Bucket ARN als Verschlüsselungskontext.

Sie können optional ein explizites Verschlüsselungskontextpaar angeben, indem Sie den `x-amz-server-side-encryption-context` Header in einer zonalen API Endpunktanforderung verwenden, z. B. [CreateSession](#) Der Wert dieses Headers ist eine Base64-kodierte Zeichenfolge mit einer Kodierung von UTF -8JSON, die den Verschlüsselungskontext in Form von Schlüssel-Wert-Paaren enthält. Für Verzeichnis-Buckets muss der Verschlüsselungskontext mit dem Standard-Verschlüsselungskontext übereinstimmen — dem Bucket Amazon Resource Name (ARN). Da der

Verschlüsselungskontext nicht verschlüsselt ist, sollten Sie außerdem sicherstellen, dass er keine vertraulichen Informationen enthält.

Sie können den Verschlüsselungskontext verwenden, um Ihre kryptografischen Vorgänge zu identifizieren und zu kategorisieren. Sie können auch den ARN Standardwert für den Verschlüsselungskontext verwenden, um relevante Anfragen in zu verfolgen AWS CloudTrail indem Sie sich ansehen, welcher Verzeichnis-Bucket mit welchem Verschlüsselungsschlüssel verwendet ARN wurde.

Wenn Sie im `requestParameters` Feld einer CloudTrail Protokolldatei SSE - KMS für einen Verzeichnis-Bucket verwenden, entspricht der Wert für den Verschlüsselungskontext dem Wert ARN des Buckets.

```
"encryptionContext": {
  "aws:s3express:arn": "arn:aws:s3::arn:aws:s3express:region:account-
id:bucket/bucket-base-name--azid--x-s3"
}
```

Außerdem gilt für die Objektverschlüsselung mit SSE - KMS in einem Verzeichnis-Bucket Ihr AWS KMS CloudTrail Ereignisse protokollieren Ihren Bucket ARN statt Ihr ObjektARN.

Senden von Anfragen für AWS KMS verschlüsselte Objekte

Auf Verzeichnis-Buckets kann nur über HTTPS (TLS) zugegriffen werden. Verzeichnis-Buckets signieren Anfragen außerdem mit AWS Signaturversion 4 (Sigv4). Weitere Informationen zum Senden von Anfragen für AWS KMS verschlüsselte Objekte finden Sie unter [Senden von Anfragen für AWS KMS verschlüsselte Objekte](#).

Wenn Ihr Objekt SSE - verwendetKMS, senden Sie keine Header für GET Verschlüsselungsanforderungen für Anfragen und HEAD Anfragen. Andernfalls erhalten Sie den Fehler HTTP400 Bad Request.

Auditing SSE — KMS Verschlüsselung in Verzeichnis-Buckets

Um die Nutzung Ihres zu überprüfen AWS KMS Schlüssel für Ihre SSE - KMS verschlüsselten Daten, die Sie verwenden können AWS CloudTrail Logs. Sie können Einblick in Ihre [kryptografischen Operationen](#) erhalten, wie zum Beispiel [GenerateDataKey](#) und [Decrypt](#). CloudTrail unterstützt zahlreiche [Attributwerte](#) zum Filtern Ihrer Suche, einschließlich Ereignisname, Benutzername und Ereignisquelle.

Themen

- [Serverseitige Verschlüsselung angeben mit AWS KMS \(SSE-KMS\) für das Hochladen neuer Objekte in Verzeichnis-Buckets](#)

Serverseitige Verschlüsselung angeben mit AWS KMS (SSE-KMS) für das Hochladen neuer Objekte in Verzeichnis-Buckets

Um Ihre Daten für Verzeichnis-Buckets mit serverseitiger Verschlüsselung zu verschlüsseln, können Sie entweder serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) (Standard) oder serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (-). SSE KMS Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanforderungen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Wenn Sie einen anderen Verschlüsselungstyp für einen Verzeichnis-Bucket angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS). Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [Von AWS verwalteter Schlüssel](#)(aws/s3) wird nicht unterstützt. Ihre KMS Konfiguration kann für die gesamte Lebensdauer des Buckets nur einen vom [Kunden verwalteten Schlüssel](#) pro SSE Verzeichnis-Bucket unterstützen. Nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, können Sie den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben. Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objekte mit SSE - angeben, müssen Sie sicherstellenKMS, dass der Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Verzeichnis-Buckets angegeben haben. Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

Sie können die Verschlüsselung anwenden, wenn Sie entweder ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren. Wenn Sie die Verschlüsselung eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen.

Sie können SSE — KMS mithilfe der REST API Operationen — angeben, AWS SDKs, und die AWS Command Line Interface (AWS CLI).

Note

- Bei Verzeichnis-Buckets ist das Verhalten beim Überschreiben der Verschlüsselung wie folgt:
 - Wenn Sie [CreateSession](#) mit der REST API zur Authentifizierung und Autorisierung von API Anfragen an zonale Endpunkte außer [CopyObject](#) und verwenden [UploadPartCopy](#), können Sie die Verschlüsselungseinstellungen auf SSE -S3 oder auf SSE — KMS nur dann überschreiben, wenn Sie zuvor die Standardverschlüsselung des Buckets mit angegeben haben. SSE KMS
 - Wenn Sie mit dem [CreateSession](#) AWS CLI oder der AWS SDKsum API Anfragen an zonale Endgeräte zu authentifizieren und zu autorisieren, mit Ausnahme von [CopyObject](#) und [UploadPartCopy](#), Sie können die Verschlüsselungseinstellungen überhaupt nicht überschreiben.
 - Wenn Sie [CopyObject](#) Anfragen stellen, können Sie die Verschlüsselungseinstellungen auf SSE -S3 oder auf SSE — KMS nur dann überschreiben, wenn Sie zuvor die Standardverschlüsselung des Buckets mit — angegeben haben. SSE KMS Wenn Sie [UploadPartCopy](#) Anfragen stellen, können Sie die Verschlüsselungseinstellungen nicht überschreiben.
- Sie können Multiregion verwenden AWS KMS keys auf Amazon S3. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter [Verwenden von Schlüsseln für mehrere Regionen in](#) AWS Key Management Service Entwicklerhandbuch.
- Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie die Erlaubnis, den Schlüssel zu verwenden. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [KMSSchlüssel erstellen, die von anderen Konten verwendet werden können](#), in AWS Key Management Service Leitfaden für Entwickler.

Mit dem REST API

Note

Pro Verzeichnis-Bucket wird für die gesamte Lebensdauer des Buckets nur ein vom [Kunden verwalteter Schlüssel](#) unterstützt. [Von AWS verwalteter Schlüssel](#)(aws/s3) wird nicht unterstützt. Nachdem Sie SSE - KMS als Standardverschlüsselungskonfiguration Ihres Buckets mit einem vom Kunden verwalteten Schlüssel angegeben haben, können Sie den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht mehr ändern.

Bei [API Vorgängen auf zonaler Endpunktebene \(Objektebene\)](#) mit Ausnahme von [CopyObject](#) und [UploadPartCopy](#) authentifizieren und autorisieren Sie Anfragen mit geringer Latenz. [CreateSession](#) Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschten Verschlüsselungskonfigurationen verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder Objektanforderungen nicht überschreiben. PUT Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln verschlüsselt und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter [Serverseitige Verschlüsselung angeben mit AWS KMS für Uploads neuer Objekte](#).

Bei API Aufrufen von zonalen Endpunkten (außer [CopyObject](#) und [UploadPartCopy](#)) mit dem REST API können Sie die Werte der Verschlüsselungseinstellungen (`x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, `x-amz-server-side-encryption-context`, und `x-amz-server-side-encryption-bucket-key-enabled`) aus der `CreateSession` Anfrage nicht überschreiben. Sie müssen diese Werte für die Verschlüsselungseinstellungen nicht explizit in API Aufrufen von zonalen Endpunkten angeben, und Amazon S3 verwendet die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage, um neue Objekte im Verzeichnis-Bucket zu schützen.

Note

Wenn Sie das verwenden AWS CLI oder das AWS SDKs, für `CreateSession`, das Sitzungstoken wird automatisch aktualisiert, um Dienstunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Das Tool AWS CLI oder der AWS SDKs verwendet die Standard-Verschlüsselungskonfiguration des Buckets für die `CreateSession` Anfrage. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen in der `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer [CopyObject](#) und [UploadPartCopy](#)) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben.

Um neue Objektkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [CopyObject](#) Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objektkopien mit SSE - angeben, müssen Sie sicherstellen KMS, dass der Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Directory-Buckets angegeben haben. Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom [Kunden verwalteten Schlüssel](#)) angeben. [UploadPartCopy](#) Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den [UploadPartCopy](#) Anforderungsheadern angeben. Außerdem müssen die Verschlüsselungseinstellungen, die Sie in der [CreateMultipartUpload](#) Anfrage angeben, mit der Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.

Themen

- [Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS](#)
- [Verschlüsselungskontext \(x-amz-server-side-encryption-context\)](#)
- [AWS KMS Schlüssel-ID \(x-amz-server-side-encryption-aws-kms-key-id\)](#)
- [S3-Bucket-Schlüssel \(x-amz-server-side-encryption-aws-bucket-key-enabled\)](#)

Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS

Die folgenden REST API Operationen auf Objektebene in Verzeichnis-Buckets akzeptieren die Header `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, und `x-amz-server-side-encryption-context` Request-Header.

- [CreateSession](#)— Wenn Sie API Operationen an zonalen Endpunkten (Objektebene) (außer CopyObject und UploadPartCopy) verwenden, können Sie diese Anforderungsheader angeben.
- [PutObject](#)— Wenn Sie Daten mithilfe der PUT API Operation hochladen, können Sie diese Anforderungsheader angeben.
- [CopyObject](#) – Wenn Sie ein Objekt kopieren, erhalten Sie ein Quell- und ein Zielobjekt. Wenn Sie bei der CopyObject Operation SSE KMS -Header übergeben, werden sie nur auf das Zielobjekt angewendet.
- [CreateMultipartUpload](#)— Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie diese Header angeben. Sie geben diese Header in der Anfrage an.
CreateMultipartUpload

Die Antwortheader der folgenden REST API Operationen geben den `x-amz-server-side-encryption` Header zurück, wenn ein Objekt mithilfe serverseitiger Verschlüsselung gespeichert wird.

- [CreateSession](#)
- [PutObject](#)
- [CopyObject](#)
- [POST Object](#)
- [CreateMultipartUpload](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [CompleteMultipartUpload](#)
- [GetObject](#)
- [HeadObject](#)

⚠ Important

- Alle GET und PUT Anfragen für ein Objekt, geschützt durch AWS KMS schlagen fehl, wenn Sie diese Anfragen nicht mithilfe von Transport Layer Security (TLS) oder Signature Version 4 stellen.
- Wenn Ihr Objekt SSE - verwendetKMS, senden Sie keine Header für GET Verschlüsselungsanfragen für Anfragen und HEAD Anfragen, da Sie sonst den BadRequest Fehler HTTP400 erhalten.

Verschlüsselungskontext (`x-amz-server-side-encryption-context`)

Wenn Sie angeben `x-amz-server-side-encryption:aws:kms`, API unterstützt Amazon S3 Sie dabei, optional einen expliziten Verschlüsselungskontext mit dem `x-amz-server-side-encryption-context` Header bereitzustellen. Bei Verzeichnis-Buckets besteht ein Verschlüsselungskontext aus einer Reihe von Schlüssel-Wert-Paaren, die kontextbezogene Informationen zu den Daten enthalten. Der Wert muss dem Standard-Verschlüsselungskontext entsprechen — dem Bucket Amazon Resource Name (ARN). Ein zusätzlicher Wert für den Verschlüsselungskontext wird nicht unterstützt.

Hinweise zum Verschlüsselungskontext in Verzeichnis-Buckets finden Sie unter [Verschlüsselungskontext](#). Allgemeine Informationen zum Verschlüsselungskontext finden Sie unter [AWS Key Management Service Konzepte — Verschlüsselungskontext](#) im AWS Key Management Service Leitfaden für Entwickler.

AWS KMS Schlüssel-ID (`x-amz-server-side-encryption-aws-kms-key-id`)

Sie können den Header `x-amz-server-side-encryption-aws-kms-key-id` verwenden, um die ID des vom Kunden verwalteten Schlüssels anzugeben, der zum Schutz der Daten verwendet wird.

Ihre KMS Konfiguration kann für die gesamte Lebensdauer des Buckets nur einen vom [Kunden verwalteten Schlüssel](#) pro SSE Verzeichnis-Bucket unterstützen. [Von AWS verwalteter Schlüssel](#)(aws/s3) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

- Sie stellen eine HeadObject API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

Hinweise zum Verschlüsselungskontext in Verzeichnis-Buckets finden Sie unter [AWS KMS keys](#).

S3-Bucket-Schlüssel (**x-amz-server-side-encryption-aws-bucket-key-enabled**)

S3-Bucket-Keys sind für GET PUT Vorgänge in einem Verzeichnis-Bucket immer aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE [CopyObject](#), [UploadPartCopy](#), der [Copy Operation in Batch-Operationen](#) oder [die import Jobs](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird. Hinweise zu den S3-Bucket-Keys in Verzeichnis-Buckets finden Sie unter [Verschlüsselungskontext](#)

Verwendung der AWS CLI

Note

Wenn Sie das verwenden AWS CLI, für `CreateSession`, das Sitzungstoken wird automatisch aktualisiert, um Dienstunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen für die `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer [CopyObject](#) und [UploadPartCopy](#)) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt.

Um das folgende Beispiel zu verwenden AWS CLI Befehle, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die Verwendung der serverseitigen Verschlüsselung mit angeben AWS KMS Schlüssel zum Verschlüsseln Ihrer Daten. Verwenden Sie dazu den `put-bucket-encryption` Befehl, um die Standardverschlüsselungskonfiguration des Directory-Buckets auf SSE - KMS (`aws:kms`) festzulegen. Fügen Sie der Anfrage insbesondere den `--server-side-encryption aws:kms` Header hinzu. Verwenden Sie das `--ssekms-key-id` *example-key-id*, um Ihren [Kunden hinzuzufügen \(verwaltet\) AWS KMS Schlüssel](#), den Sie erstellt haben. Wenn Sie angeben `--server-side-encryption aws:kms`, müssen Sie eine angeben AWS KMS Schlüssel-ID Ihres vom Kunden verwalteten Schlüssels. Directory-Buckets verwenden kein AWS verwalteter Schlüssel. Ein Beispielbefehl finden Sie unter [Verwendung der AWS CLI](#).

Wenn Sie dann ein neues Objekt mit dem folgenden Befehl hochladen, verwendet Amazon S3 die Bucket-Einstellungen für die Standardverschlüsselung, um das Objekt standardmäßig zu verschlüsseln.

```
aws s3api put-object --bucket bucket-base-name--azid--x-s3 --key example-object-key --body filepath
```

Sie müssen in Ihren Zonal Endpoint API Operations Befehle nicht `-bucket-key-enabled` explizit hinzufügen. S3-Bucket-Keys sind immer für GET PUT Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE [CopyObject](#), [UploadPartCopy](#), der [Copy Operation in Batch-Operationen](#) oder [die import Jobs](#). In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird.

Sie können ein Objekt aus einem Quell-Bucket (z. B. einem Allzweck-Bucket) in einen neuen Bucket (z. B. einen Verzeichnis-Bucket) kopieren und für die Zielobjekte die KMS Verschlüsselung verwenden SSE. Verwenden Sie dazu den `put-bucket-encryption` Befehl, um die Standardverschlüsselungskonfiguration des Ziel-Buckets (z. B. eines Verzeichnis-Buckets) auf SSE - KMS (`aws:kms`) festzulegen. Ein Beispielbefehl finden Sie unter [Verwendung der AWS CLI](#). Wenn Sie dann ein Objekt mit dem folgenden Befehl kopieren, verwendet Amazon S3 die Bucket-Einstellungen für die Standardverschlüsselung, um das Objekt standardmäßig zu verschlüsseln.

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket/example-object-key --bucket bucket-base-name--azid--x-s3 --key example-object-key
```

Verwendung der AWS SDKs

Bei der Verwendung AWS SDKs, Sie können Amazon S3 zur Nutzung anfordern AWS KMS keys für serverseitige Verschlüsselung. Die folgenden Beispiele zeigen, wie Sie SSE - KMS mit dem AWS SDKs für Java und .NET. Informationen zu anderen SDKs finden Sie unter [Beispielcode und Bibliotheken](#) auf der AWS Entwicklerzentrum.

Note

Wenn Sie das verwenden AWS SDKs, für `CreateSession`, das Sitzungstoken wird automatisch aktualisiert, um Dienstunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen für die `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer [CopyObject](#) und [UploadPartCopy](#)) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsseln KMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln verschlüsselt und entschlüsselt. Weitere Informationen zur Verwendung von AWS SDKs Informationen dazu, wie Sie die standardmäßige Verschlüsselungskonfiguration eines Verzeichnis-Buckets auf SSE - festlegen können KMS, finden Sie unter [Verwendung der AWS SDKs](#).

Important

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Weitere Informationen zum Erstellen von vom Kunden verwalteten Schlüsseln finden Sie unter [Programmieren von AWS KMS API](#) in der AWS Key Management Service Leitfaden für Entwickler.

Verschlüsselung während der Übertragung

Auf S3 Express One Zone kann nur über () zugegriffen HTTPS werden. TLS

S3 Express One Zone verwendet regionale und zonale API Endpunkte. Je nachdem, welchen Amazon S3 API S3-Betrieb Sie verwenden, ist entweder ein regionaler oder ein zonaler Endpunkt erforderlich. Sie können über einen Gateway-Endpunkt der virtuellen privaten Cloud () VPC auf zonale und regionale Endpunkte zugreifen. Für die Nutzung von Gateway-Endpunkten fallen keine zusätzlichen Gebühren an. Weitere Informationen zu regionalen und zonalen API Endpunkten finden Sie unter. [Networking für S3 Express One Zone](#)

Zusätzliche Prüfsummen

S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität mit Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) auswählen: CRC32, CRC32C, SHA -1 und -256. SHA MD5 Auf Prüfsummen basierende Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

Weitere Informationen finden Sie unter [Bewährte Methoden für zusätzliche S3-Prüfsummen](#).

Löschen von Daten

Sie können ein oder mehrere Objekte direkt aus S3 Express One Zone löschen, indem Sie die Amazon S3 S3-Konsole verwenden. AWS SDKs, AWS Command Line Interface (AWS CLI) oder Amazon S3 RESTAPI. Für alle Objekte in Ihrem S3-Bucket entstehen Speicherkosten, deshalb sollten Sie Objekte löschen, die Sie nicht mehr benötigen.

Beim Löschen eines Objekts, das in einem Verzeichnis-Bucket gespeichert ist, werden auch alle übergeordneten Verzeichnisse rekursiv gelöscht, sofern diese übergeordneten Verzeichnisse keine anderen Objekte als das Objekt enthalten, das gelöscht wird.

Note

Multi-Faktor-Authentifizierung (MFA), Löschen und S3-Versionierung werden für S3 Express One Zone nicht unterstützt.

AWS Identity and Access Management (IAM) für S3 Express One Zone

AWS Identity and Access Management (IAM) ist ein AWS-Service das hilft Administratoren, den Zugriff auf sicher zu kontrollieren AWS Ressourcen schätzen. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon S3 S3-Ressourcen in S3 Express One Zone zu verwenden. Sie können es ohne IAM zusätzliche Kosten nutzen.

Standardmäßig haben Benutzer keine Berechtigungen für Verzeichnis-Buckets und S3-Express-One-Zone-Vorgänge. Um Zugriffsberechtigungen für Verzeichnis-Buckets IAM zu gewähren, können Sie Benutzer, Gruppen oder Rollen erstellen und diesen Identitäten Berechtigungen zuweisen. Weitere Informationen IAM dazu finden Sie unter [Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch](#).

Um Zugriff zu gewähren, können Sie Ihren Benutzern, Gruppen oder Rollen auf die folgenden Weisen Berechtigungen hinzufügen:

- Benutzer und Gruppen in AWS IAM Identity Center— Erstellen Sie einen Berechtigungssatz. Folgen Sie den Anweisungen unter [Einen Berechtigungssatz erstellen](#) im AWS IAM Identity Center Benutzerleitfaden.
- Benutzer, IAM die über einen Identitätsanbieter verwaltet werden — Erstellen Sie eine Rolle für den Identitätsverbund. Folgen Sie den Anweisungen [unter Erstellen einer Rolle für einen externen Identitätsanbieter \(Federation\)](#) im IAMBenutzerhandbuch.
- IAMRollen und Benutzer — Erstellen Sie eine Rolle, die Ihr Benutzer übernehmen kann. Folgen Sie den Anweisungen [unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM Benutzer](#) im IAMBenutzerhandbuch.

Standardmäßig sind Verzeichnis-Buckets privat und sind nur für Benutzer zugänglich, denen explizit Zugriff gewährt wurde. Die Zugriffskontrollgrenze für Verzeichnis-Buckets wird nur auf Bucket-Ebene festgelegt. Im Gegensatz dazu kann die Zugriffskontrollgrenze für Allzweck-Buckets auf Bucket-, Präfix- oder Objekt-Tag-Ebene festgelegt werden. Dieser Unterschied bedeutet, dass Verzeichnis-Buckets die einzige Ressource sind, die Sie in Bucket-Richtlinien oder IAM Identitätsrichtlinien für den Zugriff auf S3 Express One Zone aufnehmen können.

Mit S3 Express One Zone authentifizieren und autorisieren Sie Anfragen zusätzlich zur IAM Autorisierung über einen neuen sitzungsbasierten Mechanismus, der vom Vorgang abgewickelt wird. `CreateSession` API Sie können mit `CreateSession` temporäre Anmeldeinformationen

anfordern, die den Zugriff auf Ihren Bucket mit geringer Latenz ermöglichen. Diese temporären Anmeldeinformationen sind auf einen bestimmten Verzeichnis-Bucket beschränkt.

Um damit arbeiten zu können `CreateSession`, empfehlen wir die Verwendung der neuesten Version von AWS SDKs oder mit dem AWS Command Line Interface (AWS CLI). Die unterstützten AWS SDKs und die AWS CLI kümmert sich in Ihrem Namen um die Einrichtung, Aktualisierung und Beendigung der Sitzung.

Sie verwenden Sitzungstoken nur für zonale Operationen (Objektebene) (außer `CopyObject` und `HeadBucket`), um die mit der Autorisierung verbundene Latenz auf eine Reihe von Anforderungen in einer Sitzung zu verteilen. Für regionale API Endpunktoperationen (Operationen auf Bucket-Ebene) verwenden Sie die IAM Autorisierung, was nicht die Verwaltung einer Sitzung beinhaltet. Weitere Informationen erhalten Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#) und [CreateSession-Autorisierung](#).

Weitere Informationen zu IAM S3 Express One Zone finden Sie in den folgenden Themen.

Themen

- [Auftraggeber](#)
- [Ressourcen](#)
- [Aktionen für S3 Express One Zone](#)
- [Bedingungsschlüssel für S3 Express One Zone](#)
- [Wie werden API Operationen autorisiert und authentifiziert](#)

Auftraggeber

Wenn Sie eine ressourcenbasierte Richtlinie erstellen, um Zugriff auf Ihre Buckets zu gewähren, müssen Sie das `Principal`-Element verwenden, um die Person oder Anwendung anzugeben, die eine Anforderung für eine Aktion oder einen Vorgang auf dieser Ressource stellen kann. Für Verzeichnis-Bucket-Richtlinien können Sie die folgenden Prinzipale verwenden:

- Importieren in `&S3;` AWS Konto
- Ein IAM Benutzer
- Eine IAM Rolle
- Ein Verbundbenutzer

Weitere Informationen finden Sie unter [Principal](#) im IAM-Benutzerhandbuch.

Ressourcen

Amazon Resource Names (ARNs) für Verzeichnis-Buckets enthalten den `s3express` Namespace, AWS-Region, der AWS Konto-ID und den Namen des Verzeichnis-Buckets, der die Availability Zone-ID enthält. Um auf Ihren Directory-Bucket zuzugreifen und Aktionen in Ihrem Directory-Bucket auszuführen, müssen Sie das folgende ARN Format verwenden:

```
arn:aws:s3express:region:account-id:bucket/base-bucket-name--azid--x-s3
```

Weitere Informationen zu finden ARNs Sie unter [Amazon Resource Names \(ARNs\)](#) im IAM-Benutzerhandbuch. Weitere Informationen zu Ressourcen finden Sie unter [IAM JSON Richtlinien](#): [Resource](#) im IAM-Benutzerhandbuch.

Aktionen für S3 Express One Zone

In einer IAM identitäts- oder ressourcenbasierten Richtlinie definieren Sie, welche S3-Aktionen zulässig oder verweigert werden. S3 Express One Zone-Aktionen entsprechen bestimmten Vorgängen. API S3 Express One Zone hat einen eindeutigen IAM Namespace, der sich vom Standard-Namespace für Amazon S3 unterscheidet. Dieser Namespace ist `s3express`.

Wenn Sie die `s3express:CreateSession` Erlaubnis erteilen, ermöglicht dies dem `CreateSession` API Vorgang, Sitzungstoken abzurufen, wenn auf Operationen auf zentraler Endpunkt-API (oder Objektebene) zugegriffen wird. Diese Sitzungstoken geben Anmeldeinformationen zurück, die verwendet werden, um Zugriff auf alle anderen zentralen API Endpunktoperationen zu gewähren. Daher müssen Sie keine Zugriffsberechtigungen für zentrale API Operationen mithilfe IAM von Richtlinien gewähren. Stattdessen ermöglicht das Sitzungstoken den Zugriff.

Weitere Informationen zu zentralen und regionalen API Endpunktoperationen finden Sie unter [Networking für S3 Express One Zone](#). Weitere Informationen zu diesem `CreateSession` API Vorgang finden Sie unter [CreateSession](#) in der Amazon Simple Storage Service API Reference.

Sie können die folgenden Aktionen im `Action` Element einer IAM Grundsatzerklärung angeben. Verwenden Sie Richtlinien, um Berechtigungen zur Ausführung eines Vorgangs zu erteilen in AWS. Wenn Sie eine Aktion in einer Richtlinie verwenden, gewähren oder verweigern Sie in der Regel den Zugriff auf den API Vorgang mit demselben Namen. In einigen Fällen steuert jedoch eine einzelne Aktion den Zugriff auf mehr als einen API Vorgang. Der Zugriff auf Aktionen auf Bucket-Ebene kann nur über IAM identitätsbasierte Richtlinien (Benutzer oder Rolle) und nicht über Bucket-Richtlinien gewährt werden.

Aktionen und Bedingungsschlüssel für S3 Express One Zone

Aktion	API	Beschreibung	Zugriffsbene	Bedingungsschlüssel
s3express:CreateBucket	CreateBucket	Gewährt die Berechtigung zum Erstellen eines neuen Buckets.	Schreibe	s3express:authType s3express:LocationName s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256
s3express:CreateSession	CreateSession	Erteilt die Berechtigung zum Erstellen eines Sitzungstokens, mit dem Zugriff auf alle zonalen API Operationen (auf Objektebene) gewährt wird, z. B., usw. PutObject GetObject	Schreibe	s3express:authType s3express:SessionMode s3express:ResourceAccount

Aktion	API	Beschreibung	Zugriffsbene	Bedingungschlüssel
				s3express :signatureversion
				s3express :signatureAge
				s3express :TlsVersion
				s3express :x-amz-content-sha256
				s3express :x-amz-server-side-encryption
				s3express :x-amz-server-side-encryption-aws-kms-key-id

Aktion	API	Beschreibung	Zugriffsbene	Bedingungschlüssel
s3express:DeleteBucket	DeleteBucket	Erteilt die Berechtigung zum Löschen des Buckets mit dem Namen. URI	Schreibe	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsbene	Bedingungschlüssel
s3express:DeleteBucketPolicy	DeleteBucketPolicy	Gewährt die Berechtigung zum Löschen der Richtlinie für einen angegebenen Bucket.	Berechtigungsverwaltung	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsebene	Bedingungschlüssel
s3express:GetBucketPolicy	GetBucketPolicy	Gewährt die Berechtigung zum Zurückgeben der Richtlinie des angegebenen Buckets.	Lesen	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsbene	Bedingungschlüssel
s3express:GetEncryptionConfiguration	GetBucketEncryption	Erteilt die Berechtigung, die Standardverschlüsselungskonfiguration eines Verzeichnis-Buckets zurückzugeben.	Lesen	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsbene	Bedingungschlüssel
s3express:ListAllMyDirectoriesBuckets	ListDirectoryBuckets	Gewährt die Berechtigung zum Auflisten aller Verzeichnis-Buckets, die dem authentifizierten Sender der Anforderung gehören.	Auflisten	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsbene	Bedingungschlüssel
s3express:PutBucketPolicy	PutBucketPolicy	Gewährt die Berechtigung zum Hinzufügen oder Ersetzen einer Bucket-Richtlinie für einen Bucket.	Berechtigungsverwaltung	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsbene	Bedingungschlüssel
s3express:PutBucketPolicy	PutBucketPolicy	Gewährt die Berechtigung zum Hinzufügen oder Ersetzen einer Bucket-Richtlinie für einen Bucket.	Berechtigungsverwaltung	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsbene	Bedingungsschlüssel
s3express:PutEncryptionConfiguration	PutBucketEncryption oder DeleteBucketEncryption	Erteilt die Berechtigung, die Verschlüsselungskonfiguration für einen Verzeichnis-Bucket festzulegen	Schreibe	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Bedingungsschlüssel für S3 Express One Zone

S3 Express One Zone definiert die folgenden Bedingungsschlüssel, die im Condition Element einer IAM Richtlinie verwendet werden können. Diese Schlüssel können Sie verwenden, um die Bedingungen zu verfeinern, unter denen die Richtlinienanweisung angewendet wird.

Bedingungsschlüssel	Beschreibung	Typ
s3express:authType	Filtert den Zugriff nach Authentifizierungsmethode. Um eingehende Anfragen auf die Verwendung einer bestimmten Authentifizierungsmethode zu beschränken, können Sie diesen optionalen Bedingungsschlüssel verwenden. Sie können diesen Bedingung	String

Bedingungsschlüssel	Beschreibung	Typ
	<p>sschlüssel beispielsweise verwenden, um zuzulassen, dass nur der HTTP Authorization Header bei der Anforderungsauthentifizierung verwendet wird.</p> <p>Zulässige Werte: REST-HEADER , REST-QUERY-STRING</p>	
<p>s3express:Location Name</p>	<p>Filtert den Zugriff auf den CreateBucket API Vorgang nach einer bestimmten Availability Zone ID (AZ-ID), usw2-az1 z. B.</p> <p>Beispielwert: usw2-az1</p>	String
<p>s3express:Resource Account</p>	<p>Filtert den Zugriff nach dem Zugriff des Ressourcenbesitzers AWS-Konto ID.</p> <p>Um den Benutzer-, Rollen- oder Anwendungszugriff auf die Verzeichnis-Buckets einzuschränken, die einer bestimmten Person gehören AWS-Konto ID, Sie können entweder den Bedingungsschlüssel <code>aws:ResourceAccount</code> oder den <code>s3express:ResourceAccount</code> Bedingungsschlüssel verwenden . Sie können diesen Bedingungsschlüssel in beiden Fällen verwenden AWS Identity and Access Management (IAM) Identitätsrichtlinien oder Virtual Private Cloud (VPC) -Endpunkt richtlinien. Sie können diesen Bedingungsschlüssel beispielsweise verwenden, um zu verhindern, dass Clients in Ihrem VPC Unternehmen auf Buckets zugreifen, die Sie nicht besitzen.</p> <p>Beispielwert: 111122223333</p>	String

Bedingungsschlüssel	Beschreibung	Typ
s3express:SessionMode	<p>Filtert den Zugriff nach der vom CreateSession API Vorgang angeforderten Berechtigung. Standardmäßig ist die Sitzung auf ReadWrite eingestellt. Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf ReadOnly zu beschränken oder den ReadWrite -Zugriff explizit zu verweigern. Weitere Informationen finden Sie unter Beispiel für Verzeichnis-Bucket-Richtlinien für S3 Express One Zone und CreateSession in der Amazon Simple Storage Service API Reference.</p> <p>Zulässige Werte: ReadWrite , ReadOnly</p>	String
s3express:signatureAge	<p>Filtert den Zugriff nach dem Alter der Anforderungssignatur in Millisekunden. Diese Bedingung gilt nur für vorsignierte Personen URLs.</p> <p>In AWS Signaturversion 4, der Signaturschlüssel ist bis zu sieben Tage gültig. Daher sind die Signaturen auch bis zu sieben Tage lang gültig. Weitere Informationen finden Sie unter Einführung in das Signieren von Anfragen in der Amazon Simple Storage API Service-Referenz. Sie können diese Bedingung verwenden, um das Alter der Unterschrift weiter einzuschränken.</p> <p>Beispielwert: 600000</p>	Numerischer Wert
s3express:signatureversion	<p>Identifiziert die Version von AWS Signatur, die Sie für authentifizierte Anfragen unterstützen möchten. Für authentifizierte Anfragen unterstützt S3 Express One Zone Signature Version 4.</p> <p>Gültiger Wert: "AWS4-HMAC-SHA256" (identifiziert Signature Version 4)</p>	String

Bedingungsschlüssel	Beschreibung	Typ
<code>s3express:TlsVersion</code>	<p>Filtert den Zugriff nach der TLS Version, die vom Client verwendet wird.</p> <p>Sie können den <code>s3:TlsVersion</code> Bedingungsschlüssel verwenden IAM, um Richtlinien für virtuelle private Cloud-Endpunkte (VPCE) oder Buckets zu schreiben, die den Benutzer- oder Anwendungszugriff auf Verzeichnis-Buckets auf der Grundlage der vom Client verwendeten TLS Version einschränken. Sie können diesen Bedingungsschlüssel auch verwenden , um Richtlinien zu schreiben, für die eine TLS Mindestversion erforderlich ist.</p> <p>Beispielwert: <code>1.3</code></p>	Numerischer Wert

Bedingungsschlüssel	Beschreibung	Typ
<p>s3express:x-amz-content-sha256</p>	<p>Filtert den Zugriff auf nicht signierte Inhalte in Ihrem Bucket.</p> <p>Sie können diesen Bedingungsschlüssel verwenden, um nicht signierte Inhalte in Ihrem Bucket zu verbieten.</p> <p>Wenn Sie Signature Version 4 verwenden, fügen Sie bei Anfragen, die den <code>Authorization</code>-Header verwenden, den <code>x-amz-content-sha256</code>-Header in die Signaturrechnung ein und setzen dann dessen Wert auf die Hash-Nutzlast.</p> <p>Sie können diesen Bedingungsschlüssel in Ihrer Bucket-Richtlinie verwenden, um alle Uploads zu verweigern, deren Nutzlasten nicht signiert sind. Beispielsweise:</p> <ul style="list-style-type: none"> • Verweigern Sie Uploads, die den <code>Authorization</code>-Header zur Authentifizierung von Anfragen verwenden, aber die Nutzdaten nicht signieren. Weitere Informationen finden Sie unter Payload in einem einzigen Chunk übertragen in der Amazon Simple Storage Service API Reference. • Verweigern Sie Uploads, die vorsigniert verwenden. URLs Vorsignierte haben URLs immer eine. <code>UNSIGNED_PAYLOAD</code> Weitere Informationen finden Sie unter Authentifizieren von Anfragen und Authentifizierungsmethoden in der Amazon Simple Storage API Service-Referenz. <p>Günstiger Wert: <code>UNSIGNED-PAYLOAD</code></p>	<p>String</p>

Bedingungsschlüssel	Beschreibung	Typ
<code>s3express:x-amz-server-side-encryption</code>	Filtert den Zugriff nach serverseitiger Verschlüsselung Zulässige Werte: "AWS256", <code>aws:kms</code>	String
<code>s3express:x-amz-server-side-encryption-aws-kms-key-id</code>	Filtert den Zugriff nach AWS KMS vom Kunden verwalteter Schlüssel für serverseitige Verschlüsselung Beispielwert: " <code>arn:aws:kms: <i>region</i>:<i>acct-id</i>:key/<i>key-id</i></code> "	ARN

Wie werden API Operationen autorisiert und authentifiziert

In der folgenden Tabelle sind Autorisierungs- und Authentifizierungsinformationen für S3 Express One API Zone-Operationen aufgeführt. Die Tabelle zeigt für jeden API Vorgang den Namen, die API IAM Aktion, den Endpunkttyp (regional oder zonal) und den Autorisierungsmechanismus (IAModer sitzungsbasiert). In dieser Tabelle wird auch angegeben, wo der kontoübergreifende Zugriff unterstützt wird. Der Zugriff auf Aktionen auf Bucket-Ebene kann nur über IAM identitätsbasierte Richtlinien (Benutzer oder Rolle) gewährt werden, nicht über Bucket-Richtlinien.

API	Endpunkttyp	IAMAktion	Kontoübergreifender Zugriff
CreateBucket	Regional	<code>s3express>CreateBucket</code>	Nein
DeleteBucket	Regional	<code>s3express>DeleteBucket</code>	Nein
ListDirectoryBuckets	Regional	<code>s3express:ListAllMyDirectoryBuckets</code>	Nein
PutBucketPolicy	Regional	<code>s3express:PutBucketPolicy</code>	Nein
GetBucketPolicy	Regional	<code>s3express:GetBucketPolicy</code>	Nein

API	Endpunkttyp	IAMAktion	Kontoübergreifender Zugriff
DeleteBucketPolicy	Regional	s3express:DeleteBucketPolicy	Nein
PutBucketEncryption	Regional	s3express:PutEncryptionConfiguration	Nein
GetBucketEncryption	Regional	s3express:GetEncryptionConfiguration	Nein
DeleteBucketEncryption	Regional	s3express:PutEncryptionConfiguration	Nein
CreateSession	Zonal	s3express:CreateSession	Ja
CopyObject	Zonal	s3express:CreateSession	Ja
DeleteObject	Zonal	s3express:CreateSession	Ja
DeleteObjects	Zonal	s3express:CreateSession	Ja
HeadObject	Zonal	s3express:CreateSession	Ja
PutObject	Zonal	s3express:CreateSession	Ja
GetObjectAttributes	Zonal	s3express:CreateSession	Ja
ListObjectsV2	Zonal	s3express:CreateSession	Ja
HeadBucket	Zonal	s3express:CreateSession	Ja
CreateMultipartUpload	Zonal	s3express:CreateSession	Ja
UploadPart	Zonal	s3express:CreateSession	Ja

API	Endpunkttyp	IAMAktion	Kontoübergreifender Zugriff
UploadPartCopy	Zonal	s3express:CreateSession	Ja
CompleteMultipartUpload	Zonal	s3express:CreateSession	Ja
AbortMultipartUpload	Zonal	s3express:CreateSession	Ja
ListParts	Zonal	s3express:CreateSession	Ja
ListMultipartUploads	Zonal	s3express:CreateSession	Ja

Auf IAM-Identitäten basierende IAM-Richtlinien für S3 Express One Zone

Bevor Sie Verzeichnis-Buckets erstellen oder die Speicherklasse Amazon S3 Express One Zone verwenden können, müssen Sie Ihrer AWS Identity and Access Management (IAM)-Rolle oder Ihren Benutzern die erforderlichen Berechtigungen erteilen. Diese Beispielrichtlinie ermöglicht den Zugriff auf den `CreateSession`-API-Vorgang (zur Verwendung mit API-Vorgängen auf zonalem Endpunkt [Objektebene]) und auf alle API-Operationen des regionalen Endpunkts (Bucket-Ebene). Diese Richtlinie ermöglicht die Verwendung des `CreateSession`-API-Vorgangs mit allen Verzeichnis-Buckets, aber die API-Operationen für regionale Endpunkte sind nur für die Verwendung mit dem angegebenen Verzeichnis-Bucket zulässig. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessRegionalEndpointAPIs",
      "Effect": "Allow",
      "Action": [
        "s3express:DeleteBucket",
```

```

        "s3express:DeleteBucketPolicy",
        "s3express:CreateBucket",
        "s3express:PutBucketPolicy",
        "s3express:GetBucketPolicy",
        "s3express:ListAllMyDirectoryBuckets"
    ],
    "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-
name--azid--x-s3/*"
  },
  {
    "Sid": "AllowCreateSession",
    "Effect": "Allow",
    "Action": "s3express:CreateSession",
    "Resource": "*"
  }
]
}

```

Beispiel für Verzeichnis-Bucket-Richtlinien für S3 Express One Zone

Dieser Abschnitt zeigt Verzeichnis-Bucket-Richtlinien zur Verwendung mit der Speicherklasse Amazon S3 Express One Zone. Wenn Sie diese Richtlinien verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

Die folgende Beispiel-Bucket-Richtlinie ermöglicht es AWS-Konto-ID **111122223333**, den `CreateSession`-API-Vorgang mit der `ReadWrite`-Standardsitzung für den angegebenen Verzeichnis-Bucket zu verwenden. Diese Richtlinie gewährt Zugriff auf die API-Operationen des zonalen Endpunkts (Objektebene).

Example – Bucket-Richtlinie, um **CreateSession**-Aufrufe mit der **ReadWrite**-Standardsitzung zuzulassen

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadWriteAccess",
      "Effect": "Allow",
      "Resource": "arn:aws:s3express:us-west-2:account-id:bucket/bucket-base-
name--azid--x-s3",

```



```

        "Principal": {
            "AWS": [
                "111122223333"
            ]
        },
        "Action": [
            "s3express:CreateSession"
        ]
    }
]
}

```

Example – Bucket-Richtlinie, um **CreateSession**-Aufrufe mit einer **ReadOnly**-Sitzung zuzulassen

Das folgende Beispiel für eine Bucket-Richtlinie ermöglicht der AWS-Konto-ID **111122223333**, den **CreateSession**-API-Vorgang zu verwenden. Diese Richtlinie verwendet den **s3express:SessionMode**-Bedingungsschlüssel mit dem **ReadOnly**-Wert, um eine schreibgeschützte Sitzung einzurichten.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnlyAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "111122223333"
      },
      "Action": "s3express:CreateSession",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "s3express:SessionMode": "ReadOnly"
        }
      }
    }
  ]
}

```

Example – Bucket-Richtlinie, um den kontoübergreifenden Zugriff für **CreateSession**-Aufrufe zuzulassen

Die folgende Beispiel-Bucket-Richtlinie ermöglicht es AWS-Konto-ID **111122223333**, den CreateSession-API-Vorgang für den angegebenen Verzeichnis-Bucket zu verwenden, dessen Eigentümer AWS-Konto-ID **444455556666** ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "111122223333"
      },
      "Action": [
        "s3express:CreateSession"
      ],
      "Resource": "arn:aws:s3express:us-west-2:444455556666:bucket/bucket-base-name--azid--x-s3"
    }
  ]
}
```

CreateSession-Autorisierung

Amazon S3 Express One Zone unterstützt sowohl die AWS Identity and Access Management (AWS IAM)-Autorisierung als auch die sitzungsbasierte Autorisierung:

- Um regionale Endpunkt-API-Operationen (Operationen auf Bucket-Ebene oder Steuerebene) mit S3 Express One Zone zu verwenden, verwenden Sie das IAM-Autorisierungsmodell, das keine Sitzungsverwaltung beinhaltet. Berechtigungen werden für Aktionen einzeln erteilt. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).
- Um API-Vorgänge für zonale Endpunkte (Vorgänge auf Objektebene oder Datenebene) zu verwenden, verwenden Sie den CreateSession-API-Vorgang, um Sitzungen zu erstellen und zu

verwalten, die für die Autorisierung von Datenanforderungen mit geringer Latenz optimiert sind. Um ein Sitzungstoken abzurufen und zu verwenden, müssen Sie die `s3express:CreateSession`-Aktion für Ihren Verzeichnis-Bucket in einer identitätsbasierten Richtlinie oder einer Bucket-Richtlinie zulassen. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#). Wenn Sie in der Amazon-S3-Konsole, über die AWS Command Line Interface (AWS CLI) oder mithilfe der AWS-SDKs auf S3 Express One Zone zugreifen, erstellt S3 Express One Zone für Sie eine Sitzung.

Wenn Sie die Amazon-S3-REST-API verwenden, können Sie dann den `CreateSessionAPI`-Vorgang verwenden, um temporäre Sicherheitsanmeldeinformationen abzurufen, die eine Zugriffsschlüssel-ID, einen geheimen Zugriffsschlüssel, ein Sitzungstoken und eine Ablaufzeit enthalten. Die temporären Anmeldeinformationen bieten die gleichen Berechtigungen wie langfristige Sicherheitsanmeldeinformationen, z. B. IAM-Benutzer-Anmeldeinformationen müssen jedoch ein Sitzungstoken beinhalten.

Sitzungsmodus

Der Sitzungsmodus definiert den Umfang der Sitzung. In Ihrer Bucket-Richtlinie können Sie den `s3express:SessionMode`-Bedingungsschlüssel angeben, um zu steuern, wer eine `ReadWrite`- oder `ReadOnly`-Sitzung erstellen kann. Weitere Informationen zu `ReadWrite`- und `ReadOnly`-Sitzungen finden Sie unter dem `x-amz-create-session-mode`-Parameter für [CreateSession](#) in der Amazon-S3-API-Referenz. Informationen dazu, wie Sie eine Bucket-Richtlinie erstellen, finden Sie unter [Beispiel für Verzeichnis-Bucket-Richtlinien für S3 Express One Zone](#).

Sitzungs-Token

Wenn Sie einen Aufruf mit temporären Sicherheitsanmeldeinformationen tätigen, muss dieser ein Sitzungstoken enthalten. Das Sitzungstoken wird zusammen mit den temporären Anmeldeinformationen zurückgegeben. Ein Sitzungstoken ist auf Ihren Verzeichnis-Bucket beschränkt und wird verwendet, um zu überprüfen, ob die Sicherheitsanmeldedaten gültig und nicht abgelaufen sind. Um Ihre Sitzungen zu schützen, laufen temporäre Sicherheitsanmeldedaten nach 5 Minuten ab.

CopyObject und HeadBucket

Temporäre Sicherheitsanmeldedaten sind auf einen bestimmten Verzeichnis-Bucket beschränkt und werden automatisch für alle API-Aufrufe von zonalen Operationen (auf Objektebene) für einen bestimmten Verzeichnis-Bucket aktiviert. Im Gegensatz zu anderen API-Vorgängen für zonale

Endpunkte verwenden CopyObject, HeadBucket und CreateSession keine Authentifizierung. Alle CopyObject- und HeadBucket-Anforderungen müssen mithilfe von IAM-Anmeldeinformationen authentifiziert und signiert werden. CopyObject und HeadBucket sind jedoch wie andere API-Vorgänge für zonale Endpunkte weiterhin von s3express:CreateSession autorisiert.

Weitere Informationen finden Sie unter [CreateSession](#) in der API-Referenz zu Amazon Simple Storage Service.

Bewährte Methoden für die Sicherheit in S3 Express One Zone

Amazon S3 Express One Zone enthält eine Reihe von Sicherheits-Features, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden stellen allgemeine Richtlinien und keine vollständige Sicherheitslösung dar. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Empfehlungen und nicht als bindend ansehen.

Standardeinstellungen für „Öffentlichen Zugriff blockieren“ und „Objekteigentümerschaft“

Um die Speicherklasse S3 Express One Zone zu verwenden, müssen Sie einen S3-Verzeichnis-Bucket verwenden. Verzeichnis-Buckets unterstützen S3 Block Public Access und S3 Object Ownership. Diese S3-Features werden für die Überwachung und Verwaltung des Zugriffs auf Ihre Buckets und Objekte verwendet.

Alle Einstellungen zum Blockieren jeglichen öffentlichen Zugriffs sind für Verzeichnis-Buckets standardmäßig aktiviert. Darüber hinaus ist Object Ownership auf Bucket Owner enforced gesetzt, was bedeutet, dass Zugriffskontrolllisten (ACLs) deaktiviert sind. Diese Einstellungen können nicht geändert werden. Weitere Informationen zu diesen Features finden Sie unter [the section called „Blockieren des öffentlichen Zugriffs“](#) und [the section called „Steuern der Objekteigentümerschaft“](#).

Note

Sie können keinen Zugriff auf Objekte gewähren, die in Verzeichnis-Buckets gespeichert sind. Sie können nur Zugriff auf Ihre Verzeichnis-Buckets gewähren. Das Autorisierungsmodell für S3 Express One Zone unterscheidet sich vom Autorisierungsmodell für Amazon S3. Weitere Informationen finden Sie unter [CreateSession-Autorisierung](#).

Authentifizierung und Autorisierung

Die Authentifizierungs- und Autorisierungsmechanismen für S3 Express One Zone unterscheiden sich je nachdem, ob Sie Anfragen an zonale API Endpunktoperationen oder regionale API Endpunktoperationen stellen. Zonale API Operationen sind Operationen auf Objektebene (Datenebene). Bei regionalen API Operationen handelt es sich um Operationen auf Bucket-Ebene (Steuerungsebene).

Mit S3 Express One Zone authentifizieren und autorisieren Sie Anfragen an den API Betrieb zentraler Endgeräte mithilfe eines neuen sitzungsbasierten Mechanismus, der für die geringste Latenz optimiert ist. Mit der sitzungsbasierten Authentifizierung AWS SDKs verwenden Sie den `CreateSession` API Vorgang, um temporäre Anmeldeinformationen anzufordern, die den Zugriff auf Ihren Verzeichnis-Bucket mit geringer Latenz ermöglichen. Diese temporären Anmeldeinformationen sind auf einen bestimmten Verzeichnis-Bucket beschränkt und laufen nach 5 Minuten ab. Sie können diese temporären Anmeldeinformationen verwenden, um zonale Aufrufe (Objektebene) API zu signieren. Weitere Informationen finden Sie unter [CreateSession-Autorisierung](#).

Signieren von Anforderungen mit S3-Express-One-Zone-Anmeldeinformationen

Sie verwenden Ihre S3 Express One Zone-Anmeldeinformationen, um Anfragen an zonale Endpunkte (Objektebene) API mit zu signieren AWS Signaturversion 4, mit `s3express` als Dienstname. Wenn Sie Ihre Anforderungen signieren, verwenden Sie den geheimen Schlüssel, der von `CreateSession` zurückgegeben wurde, und stellen auch das Sitzungstoken mit dem `x-amzn-s3session-token` header bereit. Weitere Informationen finden Sie unter [CreateSession](#).

Die [unterstützten AWS SDKs](#) für die S3 Express One Zone-Klasse verwalten Sie Anmeldeinformationen und Unterschriften in Ihrem Namen. Wir empfehlen die Verwendung von AWS SDKs für S3 Express One Zone, um Anmeldeinformationen zu aktualisieren und Anfragen für Sie zu signieren.

Anfragen mit IAM Anmeldeinformationen signieren

Alle regionalen API Aufrufe (auf Bucket-Ebene) müssen authentifiziert und signiert werden von AWS Identity and Access Management (IAM) Anmeldeinformationen statt temporärer Sitzungsanmeldedaten. IAM Anmeldeinformationen bestehen aus der Zugriffsschlüssel-ID und dem geheimen Zugriffsschlüssel für die IAM Identitäten. Alle `CopyObject` `HeadBucket` Anfragen müssen außerdem authentifiziert und mithilfe IAM von Anmeldeinformationen signiert werden.

Um die geringste Latenz für Ihre zonalen Betriebsaufrufe (auf Objektebene) zu erreichen, empfehlen wir, zum Signieren Ihrer Anfragen die S3-Express-One-Zone-Anmeldeinformationen zu verwenden, die Sie beim Aufruf von `CreateSession` erhalten haben, mit Ausnahme von Anforderungen an `CopyObject` und `HeadBucket`.

Verwenden Sie AWS CloudTrail

AWS CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service auf Amazon S3. Sie können die von gesammelten Informationen verwenden CloudTrail , um Folgendes zu ermitteln:

- Die Anforderung, die an Amazon S3 gestellt wurde
- Die IP-Adresse, von der die Anforderung erfolgt ist
- Wer die Anforderung gestellt hat
- Wann die Anforderung gestellt wurde
- Zusätzliche Details zur Anforderung

Wenn Sie Ihre einrichten AWS-Konto, CloudTrail Verwaltungsereignisse sind standardmäßig aktiviert. Die folgenden regionalen API Endpunktoperationen (Operationen auf Bucket-Ebene oder Kontrollebene) werden protokolliert. API CloudTrail

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [PutBucketPolicy](#)
- [GetBucketPolicy](#)
- [ListDirectoryBuckets](#)
- [ListMultipartUploads](#)
- [PutBucketEncryption](#)
- [GetBucketEncryption](#)
- [DeleteBucketEncryption](#)

Note

`ListMultipartUploads` ist ein Vorgang mit zonalem Endpunkt. API Es wird jedoch CloudTrail als Verwaltungsereignis protokolliert. Weitere Informationen finden Sie unter [ListMultipartUploads](#) in der Amazon Simple Storage Service API Reference.

Standardmäßig protokollieren CloudTrail Trails keine Datenereignisse, aber Sie können Trails so konfigurieren, dass Datenereignisse für von Ihnen angegebene Verzeichnis-Buckets oder Datenereignisse für alle Verzeichnis-Buckets in Ihrem AWS Konto. Die folgenden zonalen API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) werden protokolliertAPI. CloudTrail

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CreateSession](#)
- [CopyObject](#)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [GetObject](#)
- [GetObjectAttributes](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListObjectsV2](#)
- [ListParts](#)
- [PutObject](#)
- [UploadPart](#)
- [UploadPartCopy](#)

Weitere Informationen zur Verwendung von AWS CloudTrail mit S3 Express One Zone, siehe [Protokollierung mit AWS CloudTrail für S3 Express One Zone](#).

Implementieren Sie die Überwachung mithilfe von AWS Überwachungstools

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Sicherheit, Verfügbarkeit und Leistung von Amazon S3 und Ihrer AWS Lösungen. AWS bietet verschiedene Tools und Dienste, mit denen Sie Amazon S3 und Ihre anderen überwachen können AWS-Services. Sie können beispielsweise CloudWatch Amazon-Metriken für Amazon S3 überwachen, insbesondere die `BucketSizeBytes` und `NumberOfObjects` Speichermetriken.

Objekte, die in der Speicherklasse S3 Express One Zone gespeichert sind, werden in den Speichermetriken `BucketSizeBytes` und `NumberOfObjects` für Amazon S3 nicht berücksichtigt. Die Speichermetriken `BucketSizeBytes` und `NumberOfObjects` werden jedoch für S3 Express One Zone unterstützt. Um die Metriken Ihrer Wahl zu sehen, können Sie zwischen den Amazon-S3-Speicherklassen und der Speicherklasse S3 Express One Zone unterscheiden, indem Sie eine `StorageType`-Dimension angeben. Weitere Informationen finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

Weitere Informationen erhalten Sie unter [Metriken mit Amazon überwachen CloudWatch](#) und [Überwachen von Amazon S3](#).

Protokollierung mit AWS CloudTrail für S3 Express One Zone


Amazon S3 ist integriert mit AWS CloudTrail, ein Service, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst. CloudTrail erfasst alle API Aufrufe für Amazon S3 als Ereignisse. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon S3 gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln. Wenn eine unterstützte Ereignisaktivität in Amazon S3 auftritt, wird diese Aktivität in einem CloudTrail Ereignis aufgezeichnet. Sie können Folgendes verwenden ... AWS CloudTrail Pfad, um Verwaltungsereignisse und Datenereignisse für S3 Express One Zone zu protokollieren. Weitere Informationen finden Sie unter [Amazon S3 CloudTrail S3-Ereignisse](#) und [Was ist AWS CloudTrail?](#) in der AWS CloudTrail Benutzerleitfaden.

CloudTrail Verwaltungsereignisse für S3 Express One Zone

Standardmäßig werden Aktionen auf Bucket-Ebene für Verzeichnis-Buckets als Verwaltungsereignisse CloudTrail protokolliert. Die `eventsource` für CloudTrail Verwaltungsereignisse für S3 Express One Zone ist `s3express.amazonaws.com`. Wenn Sie Ihre einrichten AWS Konto, CloudTrail Verwaltungsereignisse sind standardmäßig aktiviert. Die folgenden

regionalen API Endpunktoperationen (Operationen auf Bucket-Ebene oder Kontrollebene) werden protokolliert. API CloudTrail

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [PutBucketPolicy](#)
- [GetBucketPolicy](#)
- [ListDirectoryBuckets](#)
- [ListMultipartUploads](#)
- [GetBucketEncryption](#)
- [PutBucketEncryption](#)
- [DeleteBucketEncryption](#)

 Note

[ListMultipartUploads](#) ist ein Vorgang mit zonalem Endpunkt. API Dieser API Vorgang wird jedoch CloudTrail als Verwaltungsereignis protokolliert. Weitere Informationen finden Sie [ListMultipartUploads](#) in der Amazon Simple Storage Service API Reference.

Weitere Informationen zu CloudTrail Verwaltungsereignissen finden Sie unter [Protokollieren von Verwaltungsereignissen](#) im AWS CloudTrail Benutzerleitfaden.

CloudTrail Datenereignisse für S3 Express One Zone

Datenereignisse liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. Lesen oder Schreiben in ein Amazon-S3-Objekt). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Standardmäßig protokollieren CloudTrail Trails keine Datenereignisse, aber Sie können Trails so konfigurieren, dass Datenereignisse für Objekte protokolliert werden, die in Allzweck-Buckets und Verzeichnis-Buckets gespeichert sind. Weitere Informationen finden Sie unter [Aktivieren der Protokollierung für Objekte in einem Bucket mithilfe der Konsole](#).

Wenn Sie Datenereignisse für einen Trail in protokollieren CloudTrail, können Sie wählen, ob Sie erweiterte oder einfache Event-Selektoren verwenden möchten. Um Datenereignisse für Objekte zu

protokollieren, die in Verzeichnis-Buckets gespeichert sind, müssen Sie erweiterte Ereignisselectoren verwenden. Bei der Konfiguration erweiterter Ressourcenselectoren wählen oder spezifizieren Sie den Ressourcentyp für S3 Express One Zone, d. h. `AWS::S3Express::Object`

Die folgenden zonalen API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) werden protokolliertAPI. CloudTrail

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CreateSession](#)
- [CopyObject](#)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [GetObject](#)
- [GetObjectAttributes](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListObjectsV2](#)
- [ListParts](#)
- [PutObject](#)
- [UploadPart](#)
- [UploadPartCopy](#)

Weitere Informationen zu Datenereignissen finden Sie unter [Protokollieren](#) von CloudTrail Datenereignissen im AWS CloudTrail Benutzerleitfaden.

Weitere Informationen zu CloudTrail Ereignissen für S3 Express One Zone finden Sie in den folgenden Themen:

Themen

- [CloudTrail Beispiele für Protokolldateien für S3 Express One Zone](#)

CloudTrail Beispiele für Protokolldateien für S3 Express One Zone

Eine CloudTrail Protokolldatei enthält Informationen über den angeforderten API Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. Dieses Thema enthält Beispiele für CloudTrail Datenereignisse und Verwaltungsereignisse für S3 Express One Zone.

Themen

- [CloudTrail Beispiele für Datenereignisprotokolldateien für Amazon S3 Express One Zone](#)

CloudTrail Beispiele für Datenereignisprotokolldateien für Amazon S3 Express One Zone

Das folgende Beispiel zeigt ein Beispiel für eine CloudTrail Protokolldatei, das dies veranschaulicht [CreateSession](#).

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAI DPPEZS35WEXAMPLE:AssumedRoleSessionName",
    "arn": "arn:aws:sts::111122223333assumed-role/RoleToBeAssumed/MySessionName",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAI DPPEZS35WEXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/RoleToBeAssumed",
        "accountId": "111122223333",
        "userName": "RoleToBeAssumed"
      },
      "attributes": {
        "creationDate": "2024-07-02T00:21:16Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2024-07-02T00:22:11Z",
  "eventSource": "s3express.amazonaws.com",
  "eventName": "CreateSession",
```

```

    "awsRegion": "us-west-2",
    "sourceIPAddress": "72.21.198.68",
    "userAgent": "aws-sdk-java/2.20.160-SNAPSHOT
Linux/5.10.216-225.855.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.23+9-LTS
Java/11.0.23 vendor/Amazon.com_Inc. md/internal exec-env/AWS_Lambda_java11 io/sync
http/Apache cfg/retry-mode/standard",
    "requestParameters": {
        "bucketName": "bucket-base-name--usw2-az1--x-s3".
        "host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
        "x-amz-create-session-mode": "ReadWrite"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "AKIAI44QH8DHBEXAMPLE"
            "expiration": "Mar 20, 2024, 11:16:09 PM",
            "sessionToken": "<session token string>"
        },
    },
    "additionalEventData": {
        "SignatureVersion": "SigV4",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "bytesTransferredIn": 0,
        "AuthenticationMethod": "AuthHeader",
        "xAmzId2": "q6xhNJYmhg",
        "bytesTransferredOut": 1815,
        "availabilityZone": "usw2-az1"
    },
    "requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
    "eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
    "readOnly": true,
    "resources": [
        {
            "type": "AWS::S3Express::Object",
            "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
        },
        {
            "accountId": "111122223333"
            "type": "AWS::S3Express::DirectoryBucket",
            "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
        }
    ],

```

```

    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_128_GCM_SHA256",
      "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
    }
  }
}

```

Um zonale API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) zu verwenden, können Sie den `CreateSession` API Vorgang verwenden, um Sitzungen zu erstellen und zu verwalten, die für die Autorisierung von Datenanforderungen mit geringer Latenz optimiert sind. Sie können ihn auch verwenden, um den Umfang der `CreateSession` Protokollierung zu reduzieren. Um zu ermitteln, welche zonalen API Operationen während einer Sitzung ausgeführt wurden, können Sie die Werte `accessKeyId` unter dem `responseElements` in Ihrer `CreateSession` Protokolldatei mit denen `accessKeyId` in der Protokolldatei anderer zonaler API Operationen abgleichen. [Weitere Informationen finden Sie unter `CreateSession` Autorisierung.](#)

Das folgende Beispiel zeigt ein Beispiel für eine CloudTrail Protokolldatei, die den [GetObject](#) API Vorgang veranschaulicht, der von `CreateSession` authentifiziert wurde.

```

{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAI DPPEZS35WEXAMPLE:AssumedRoleSessionName",
    "arn": "arn:aws:sts::111122223333:assumed-role/RoleToBeAssumed/MySessionName",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "attributes": {
        "creationDate": "2024-07-02T00:21:49Z"
      }
    }
  },
  "eventTime": "2024-07-02T00:22:01Z",
  "eventSource": "s3express.amazonaws.com",
  "eventName": "GetObject",
  "awsRegion": "us-west-2",

```

```
"sourceIPAddress": "72.21.198.68",
"userAgent": "aws-sdk-java/2.25.66 Linux/5.10.216-225.855.amzn2.x86_64
OpenJDK_64-Bit_Server_VM/17.0.11+9-LTS Java/17.0.11 vendor/Amazon.com_Inc. md/internal
exec-env/AWS_Lambda_java17 io/sync http/Apache cfg/retry-mode/legacy",
"requestParameters": {
  "bucketName": "bucket-base-name--usw2-az1--x-s3",
  "x-amz-checksum-mode": "ENABLED",
  "Host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
  "key": "test-get-obj-with-checksum"
},
"responseElements": null,
"additionalEventData": {
  "SignatureVersion": "Sigv4",
  "CipherSuite": "TLS_AES_128_GCM_SHA256",
  "bytesTransferredIn": 0,
  "AuthenticationMethod": "AuthHeader",
  "x-amz-id-2": "o0y6w8K7LFsyFN",
  "bytesTransferredOut": 9,
  "availabilityZone": "usw2-az1",
  "sessionModeApplied": "ReadWrite"
},
"requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
"eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
"readOnly": true,
"resources": [
  {
    "type": "AWS::S3Express::Object",
    "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
  },
  {
    "accountId": "111122223333",
    "type": "AWS::S3Express::DirectoryBucket",
    "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
  }
],
"eventType": "AwsApiCall",
"managementEvent": false,
"recipientAccountId": "111122223333",
"eventCategory": "Data",
"tlsDetails": {
  "tlsVersion": "TLSv1.3",
```

```
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
    }
}
```

Im obigen Beispiel für eine GetObject Protokolldatei entspricht `accessKeyId` (AKIAI44QH8DHBEXAMPLE) dem Wert `accessKeyId` unter dem `responseElements` im Beispiel der CreateSession Protokolldatei. Die Übereinstimmung `accessKeyId` gibt die Sitzung an, in der der GetObject Vorgang ausgeführt wurde.

Optimieren der Leistung von Amazon S3 Express One Zone

Amazon S3 Express One Zone ist eine leistungsstarke S3-Speicherklasse mit einer einzigen Availability Zone (AZ), die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die erste S3-Speicherklasse, mit der Sie leistungsstarke Objektspeicher- und AWS-Computingressourcen wie Amazon Elastic Compute Cloud, Amazon Elastic Kubernetes Service und Amazon Elastic Container Service innerhalb einer einzigen Availability Zone zusammenführen können. Die gemeinsame Nutzung Ihrer Speicher- und Computingressourcen optimiert die Rechenleistung und die Kosten und sorgt für eine höhere Datenverarbeitungsgeschwindigkeit.

S3 Express One Zone bietet eine ähnliche Leistungselastizität wie andere S3-Speicherklassen, jedoch mit konsistenten Latenzen für Lese- und Schreibanfragen im ersten Byte im einstelligen Millisekundenbereich – bis zu zehnmal schneller als S3 Standard. S3 Express One Zone wurde von Grund auf für die Unterstützung von Burst-Durchsätzen bis hin zu sehr hohen Aggregatwerten konzipiert. Die Speicherklasse S3 Express One Zone verwendet eine maßgeschneiderte Architektur, um die Leistung zu optimieren und durch die Speicherung von Daten auf Hochleistungshardware eine konstant niedrige Anforderungslatenz zu gewährleisten. Das Objektprotokoll für S3 Express One Zone wurde verbessert, um die Authentifizierung und den Metadaten-Overhead zu optimieren.

Um die Zugriffsgeschwindigkeit weiter zu erhöhen und Hunderttausende von Anfragen pro Sekunde zu unterstützen, speichert S3 Express One Zone Daten außerdem in einem neuen Bucket-Typ, einem Amazon-S3-Verzeichnis-Bucket. Jeder S3-Verzeichnis-Bucket kann Hunderttausende von Transaktionen pro Sekunde (TPS) unterstützen.

Die Kombination aus leistungsstarker, speziell entwickelter Hardware und Software, die Datenzugriffsgeschwindigkeiten im einstelligen Millisekundenbereich bietet, und Verzeichnis-Buckets,

die für eine große Anzahl von Transaktionen pro Sekunde skalierbar sind, machen S3 Express One Zone zur besten Amazon-S3-Speicherklasse für anforderungsintensive Operationen oder leistungskritische Anwendungen.

Die folgenden Themen beschreiben bewährte Verfahren und Designmuster zur Optimierung der Leistung von Anwendungen, die die Speicherklasse S3 Express One Zone verwenden.

Themen

- [Leistungsrichtlinien und Entwurfsmuster für S3 Express One Zone](#)

Leistungsrichtlinien und Entwurfsmuster für S3 Express One Zone

Bei der Entwicklung von Anwendungen, die Objekte zu Amazon S3 Express One Zone hochladen und daraus abrufen, sollten Sie unsere bewährten Methoden befolgen, um die Leistung zu optimieren. Um die Speicherklasse S3 Express One Zone zu verwenden, müssen Sie einen S3-Verzeichnis-Bucket erstellen. Die Speicherklasse S3 Express One Zone wird für die Verwendung mit S3-Allzweck-Buckets nicht unterstützt.

Leistungsrichtlinien für alle anderen Amazon-S3-Speicherklassen und S3-Allzweck-Buckets finden Sie unter [Bewährte Methoden für Designmuster: Optimieren der Leistung von Amazon S3](#).

Zur Erzielung der besten Leistung für Ihre Anwendung bei Verwendung der Speicherklasse S3 Express One Zone und von Verzeichnis-Buckets empfehlen wir die folgenden Vorgehensweisen und Designmuster.

Themen

- [Platzieren Sie den S3-Express-One-Zone-Speicher gemeinsam mit Ihren AWS-Computingressourcen](#)
- [Verzeichnis-Buckets](#)
- [Parallelisierung horizontaler Skalierungsanforderungen für Verzeichnis-Buckets](#)
- [Verwenden der sitzungsbasierten Authentifizierung](#)
- [Bewährte Methoden für zusätzliche S3-Prüfsummen](#)
- [Verwenden der neuesten Version der AWS-SDKs und der gemeinsamen Laufzeitbibliotheken](#)
- [Fehlerbehebung bei der Leistung](#)

Platzieren Sie den S3-Express-One-Zone-Speicher gemeinsam mit Ihren AWS-Computingressourcen

Jeder Verzeichnis-Bucket wird in einer einzigen Availability Zone gespeichert, die Sie bei dessen Erstellung auswählen. Sie können damit beginnen, indem Sie neuen Verzeichnis-Bucket in einer Availability Zone erstellen, die für Ihre Computng-Workloads oder Ressourcen lokal ist. Sie können dann sofort mit Lese- und Schreibvorgängen mit sehr niedriger Latenz beginnen. Verzeichnis-Buckets sind die ersten S3-Buckets, in denen Sie die Availability Zone in einer AWS-Region auswählen können, um die Latenz zwischen Rechenleistung und Speicher zu reduzieren.

Wenn Sie in verschiedenen Availability Zones auf Verzeichnis-Buckets zugreifen, erhöht sich die Latenz. Um die Leistung zu optimieren, empfehlen wir, dass Sie von Amazon Elastic Container Service-, Amazon Elastic Kubernetes Service- und Amazon Elastic Compute Cloud-Instances, die sich nach Möglichkeit in derselben Availability Zone befinden, auf einen Directory-Bucket zugreifen.

Verzeichnis-Buckets

Jeder Verzeichnis-Bucket kann Hunderttausende von Transaktionen pro Sekunde (TPS) unterstützen. Im Gegensatz zu Allzweck-Buckets organisieren Verzeichnis-Buckets Schlüssel hierarchisch in Verzeichnissen statt nach Präfixen. Ein Präfix ist eine Zeichenfolge am Anfang des Objektschlüsselnamens. Sie können sich Präfixe als eine Möglichkeit vorstellen, Ihre Daten ähnlich wie Verzeichnisse zu organisieren. Präfixe sind jedoch keine Verzeichnisse.

Präfixe organisieren Daten in einem flachen Namespace innerhalb von Allzweck-Buckets und die Anzahl der Präfixe in einem Allzweck-Bucket ist unbegrenzt. Jedes Präfix kann mindestens 3 500 PUT/POST/DELETE oder 5 500 GET/HEAD-Anforderungen pro Sekunde erreichen. Sie können Anfragen auch über mehrere Präfixe parallelisieren, um die Leistung zu skalieren. Die Skalierung erfolgt sowohl bei Lese- als auch bei Schreiboperationen schrittweise und nicht sofort. Während Allzweck-Buckets auf Ihre neue höhere Anforderungsrate skaliert werden, erhalten Sie möglicherweise einige HTTP-Statuscode 503 (Service Unavailable)-Fehler.

Bei einem hierarchischen Namespace ist das Trennzeichen im Objektschlüssel wichtig. Das einzige unterstützte Trennzeichen ist der Schrägstrich (/). Verzeichnisse werden durch Trennzeichengrenzen bestimmt. Beispielsweise führt der Objektschlüssel `dir1/dir2/file1.txt` dazu, dass die Verzeichnisse `dir1/` und `dir2/` automatisch erstellt werden und das Objekt `file1.txt` dem `/dir2/`-Verzeichnis im Pfad `dir1/dir2/file1.txt` hinzugefügt wird.

Die Verzeichnisse, die erstellt werden, wenn Objekte in Verzeichnis-Buckets hochgeladen werden, haben keine TPS-Beschränkungen pro Präfix und werden automatisch vorskaliert, um die

Wahrscheinlichkeit von HTTP 503-Fehlern (Service Unavailable) zu verringern. Diese automatische Skalierung ermöglicht Ihren Anwendungen, Lese- und Schreib Anforderungen innerhalb von Verzeichnissen und zwischen Verzeichnissen nach Bedarf zu parallelisieren.

Parallelisierung horizontaler Skalierungsanforderungen für Verzeichnis-Buckets

Sie erreichen die beste Leistung durch die Ausgabe mehrerer gleichzeitiger Anfragen an Verzeichnis-Buckets zur Verteilung Ihrer Anforderungen auf separate Verbindungen und zur Maximierung der verfügbaren Bandbreite. S3Express One Zone hat keine Beschränkungen für die Anzahl der Verbindungen, die mit Ihrem Verzeichnis-Bucket hergestellt werden. Einzelne Verzeichnisse können die Leistung horizontal und automatisch skalieren, wenn eine große Anzahl gleichzeitiger Schreibvorgänge in dasselbe Verzeichnis stattfindet.

Wenn ein Objektschlüssel zum ersten Mal erstellt wird und sein Schlüsselname ein Verzeichnis enthält, wird das Verzeichnis automatisch für das Objekt erstellt. Bei nachfolgenden Objekt-Uploads in dasselbe Verzeichnis muss das Verzeichnis nicht erstellt werden, wodurch die Latenz beim Hochladen von Objekten in bestehende Verzeichnisse reduziert wird.

Obwohl sowohl flache als auch tiefe Verzeichnisstrukturen für das Speichern von Objekten in einem Verzeichnis-Bucket unterstützt werden, werden Verzeichnis-Buckets automatisch horizontal skaliert, wodurch die Latenz bei gleichzeitigen Uploads in dasselbe Verzeichnis oder in parallele „Verzeichnisgeschwister“ geringer ist.

Verwenden der sitzungsbasierten Authentifizierung

S3 Express One Zone und Verzeichnis-Buckets unterstützen einen neuen sitzungsbasierten Autorisierungsmechanismus zur Authentifizierung und Autorisierung von Anforderungen an einen Verzeichnis-Bucket. Bei der sitzungsbasierten Authentifizierung verwenden die AWS-SDKs automatisch den `CreateSession`-API-Vorgang, um ein temporäres Sitzungstoken zu erstellen, das für die Autorisierung von Datenanforderungen an einen Verzeichnis-Bucket mit geringer Latenz verwendet werden kann.

Die AWS-SDKs verwenden den `CreateSession`-API-Vorgang, um temporäre Anmeldeinformationen anzufordern. Anschließend erstellen und aktualisieren sie für Sie automatisch alle 5 Minuten Token. Um die Leistungsvorteile der Speicherklasse S3 Express One Zone zu nutzen, wird empfohlen, die AWS-SDKs zum Initiieren und Verwalten der `CreateSession`-API-Anforderung zu verwenden. Weitere Informationen zu diesem sitzungsbasierten Modell finden Sie unter [CreateSession-Autorisierung](#).

Bewährte Methoden für zusätzliche S3-Prüfsummen

S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Secure Hash Algorithms (SHA)- oder Cyclic Redundancy Check (CRC)-Algorithmen zur Überprüfung der Datenintegrität auswählen: CRC32, CRC32C, SHA-1 und SHA-256. MD5-based Prüfsummen werden von der Speicherklasse S3 Express One Zone nicht unterstützt.

CRC32 ist die Standardprüfsumme, die von den AWS-SDKs bei der Übertragung von Daten zu oder von S3 Express One Zone verwendet wird. Wir empfehlen die Verwendung von CRC32 und CRC32C, um die beste Leistung mit der Speicherklasse S3 Express One Zone zu erzielen.

Verwenden der neuesten Version der AWS-SDKs und der gemeinsamen Laufzeitbibliotheken

Einige der AWS-SDKs stellen auch die AWS Common Runtime (CRT)-Bibliotheken bereit, um die Leistung in S3-Clients weiter zu beschleunigen. Zu diesen SDKs gehören AWS SDK for Java 2.x, AWS SDK for C++ und AWS SDK for Python (Boto3). Der CRT-basierte S3-Client überträgt Objekte zu und von S3 Express One Zone mit verbesserter Leistung und Zuverlässigkeit, indem er automatisch den mehrteiligen Upload-API-Vorgang und Abrufe im Bytebereich verwendet, um horizontal skalierende Verbindungen zu automatisieren.

Um die höchste Leistung mit der Speicherklasse S3 Express One Zone zu erzielen, empfehlen wir, die neueste Version der AWS-SDKs zu verwenden, die die CRT-Bibliotheken enthalten, oder die AWS Command Line Interface (AWS CLI) zu verwenden.

Fehlerbehebung bei der Leistung

Wiederholung von Anforderungen für latenzsensitive Anwendungen

S3 Express One Zone wurde speziell für gleichbleibende Leistung ohne zusätzliche Anpassungen entwickelt. Die Festlegung aggressiver Timeout-Werte und Wiederholungsversuche kann jedoch weiter zu einer gleichbleibenden Latenz und Leistung beitragen. Die AWS-SDKs verfügen über konfigurierbare Timeout- und Wiederholungsversuch-Werte, die Sie an die Toleranzen Ihrer spezifischen Anwendung anpassen können.

AWS Common Runtime (CRT)-Bibliotheken und Kopplung von Amazon-EC2-Instance-Typen

Anwendungen, die eine große Anzahl von Lese- und Schreibvorgängen ausführen, benötigen wahrscheinlich mehr Arbeitsspeicher oder Computing-Kapazitäten als Anwendungen, bei denen

dies nicht der Fall ist. Wählen Sie beim Starten Ihrer Amazon-Elastic-Compute-Cloud (Amazon EC2)-Instances für Ihre leistungsfordernde Workloads die Instance-Typen aus, die über die Menge dieser Ressourcen verfügen, die Ihre Anwendung benötigt. Die S3-Express-One-Zone-Hochleistungsspeicherung lässt sich ideal mit größeren und neueren Instance-Typen mit größerem Systemspeicher und leistungsstärkeren CPUs und GPUs kombinieren, die die Vorteile von leistungsfähigerem Speicher nutzen können. Wir empfehlen außerdem, die neuesten Versionen der CRT-fähigen AWS-SDKs zu verwenden, die Lese- und Schreibanforderungen besser parallel beschleunigen können.

Verwenden Sie die sitzungsbasierte Authentifizierung in AWS-SDKs anstelle der HTTP-REST-APIs

Mit Amazon S3 können Sie auch die Leistung optimieren, wenn Sie HTTP-REST-API-Anfragen verwenden, indem Sie dieselben bewährten Methoden befolgen, die Teil der AWS-SDKs sind. Angesichts des sitzungsbasierten Autorisierungs- und Authentifizierungsmechanismus, der von S3 Express One Zone verwendet wird, empfehlen wir Ihnen jedoch dringend, die AWS-SDKs zur Verwaltung von `CreateSession` und das zugehörige verwaltete Sitzungstoken zu verwenden. Die AWS-SDKs erstellen und aktualisieren mithilfe der `CreateSession`-API-Operation automatisch Token in Ihrem Namen. Die Verwendung von `CreateSession` ermöglicht Einsparungen bei der Round-Trip-Latenz pro Anforderung an AWS Identity and Access Management (IAM) für die Autorisierung der einzelnen Anforderungen.

Entwicklung mit S3 Express One Zone

Amazon S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher zusammen mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Mit der Speicherklasse S3 Express One Zone verwenden Sie S3-Verzeichnis-Buckets zum Speichern Ihrer Daten. Jeder Verzeichnis-Bucket verwendet die Speicherklasse S3 Express One Zone, um Objekte in einer einzigen Availability Zone zu speichern, die Sie bei der Erstellung des Buckets auswählen können.

Nachdem Sie Ihren Verzeichnis-Bucket erstellt haben, können Sie sofort mit Lese- und Schreibvorgängen mit sehr niedriger Latenz beginnen. Sie können mit Ihrem Verzeichnis-Bucket kommunizieren, indem Sie eine Endpunktverbindung über eine virtuelle private Cloud (VPC) verwenden, oder Sie können zonale und regionale API Operationen verwenden, um Ihre Objekte und Verzeichnis-Buckets zu verwalten. Sie können die Speicherklasse S3 Express One Zone auch über die Amazon S3-Konsole AWS Command Line Interface (AWS CLI) und Amazon S3 verwenden RESTAPI. AWS SDKs

Die Speicherklasse Amazon S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das [Amazon S3 Service Level Agreement](#) unterstützt. Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone wurde entwickelt, um mit gleichzeitigen Geräteausfällen umzugehen, indem verlorene Redundanz schnell erkannt und behoben werden kann. Wenn das vorhandene Gerät ausfällt, leitet S3 Express One Zone Anfragen automatisch an neue Geräte innerhalb einer Availability Zone weiter. Diese Redundanz trägt dazu bei, den unterbrechungsfreien Zugriff auf Ihre Daten innerhalb einer Availability Zone sicherzustellen.

Themen

- [Availability Zones und Regionen bei S3 Express One Zone](#)
- [Regionale und zonale Endpunkte](#)
- [Arbeiten mit S3 Express One Zone mithilfe der S3-Konsole AWS CLI, und AWS SDKs](#)
- [S3 Express API One-Zone-Betrieb](#)

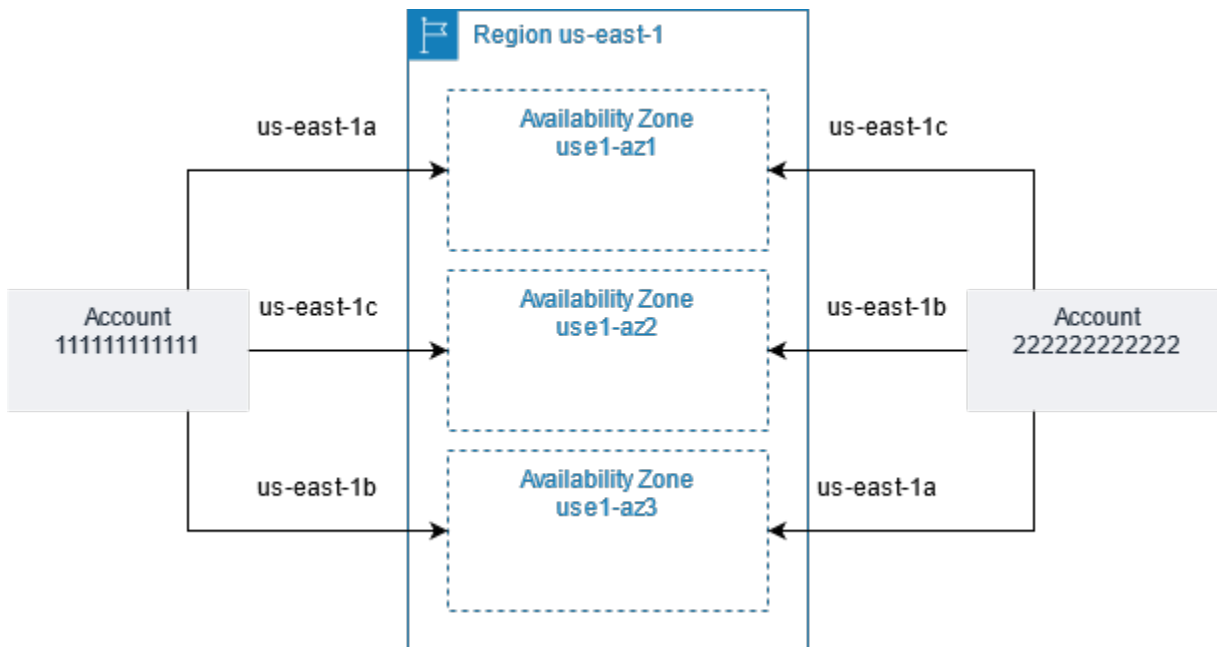
Availability Zones und Regionen bei S3 Express One Zone

Eine Availability Zone ist eines oder mehrere diskrete Rechenzentren mit redundanter Stromversorgung, Vernetzung und Konnektivität in einem AWS-Region. Um Abrufe mit niedriger Latenz zu optimieren, werden Objekte in der Speicherklasse Amazon S3 Express One Zone redundant in S3-Verzeichnis-Buckets in einer einzigen Availability Zone gespeichert, die für Ihr Computing-Workload lokal ist. Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie die Availability Zone und den AWS-Region Speicherort Ihres Buckets aus.

AWS ordnet die physischen Availability Zones nach dem Zufallsprinzip den jeweiligen Availability Zonen-Namen zu AWS-Konto. Dieser Ansatz hilft dabei, Ressourcen auf die Availability Zones in einer Region zu verteilen AWS-Region, anstatt dass sich die Ressourcen wahrscheinlich auf die erste Availability Zone für jede Region konzentrieren. Daher entspricht die Availability Zone us-east-1a für Sie AWS-Konto möglicherweise nicht demselben physischen Standort wie us-east-1a für eine andere AWS-Konto. Weitere Informationen finden Sie unter [Regionen und Availability Zones](#) im Amazon EC2 EC2-Benutzerhandbuch.

Um die Availability Zones kontenübergreifend zu koordinieren, müssen Sie die AZ-ID verwenden, die eine eindeutige und konsistente Kennung für eine Availability Zone ist. use1-az1 ist beispielsweise eine AZ-ID für die us-east-1 Region und sie hat in jeder AWS-Konto Region denselben physischen

Standort. Die folgende Abbildung zeigt, dass die AZ-IDs für jedes Konto identisch sind, auch wenn die Namen der Availability Zone für jedes Konto unterschiedlich zugeordnet werden können.



Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das [Amazon S3 Service Level Agreement](#) unterstützt. Weitere Informationen finden Sie unter [Einzelne Availability Zone](#).

S3 Express One Zone wird in den folgenden Regionen und Availability Zones unterstützt:

Von S3 Express One Zone unterstützte Regionen und Availability Zones

Name der Region	Regionscode	Availability Zone-ID
USA Ost (Nord-Virginia)	us-east-1	use1-az4
		use1-az5
		use1-az6
USA West (Oregon)	us-west-2	usw2-az1
		usw2-az3
		usw2-az4

Name der Region	Regionscode	Availability Zone-ID
Asien-Pazifik (Tokio)	ap-northeast-1	apne1-az1
		apne1-az4
Europa (Stockholm)	eu-north-1	eun1-az1
		eun1-az2
		eun1-az3

Regionale und zonale Endpunkte

Um von Ihrer Virtual Private Cloud (VPC) aus auf die regionalen und zonalen Endpunkte für Amazon S3 Express One Zone zuzugreifen, können Sie Gateway-VPC-Endpunkte verwenden. Nachdem Sie den Gateway-Endpunkt erstellt haben, können Sie ihn als Ziel in Ihrer Routing-Tabelle für Datenverkehr hinzufügen, der von Ihrer VPC zu S3 Express One Zone bestimmt ist. Für die Nutzung von Gateway-Endpunkten fallen keine zusätzlichen Gebühren an. Weitere Informationen dazu, wie Sie Gateway-VPC-Endpunkte konfigurieren, finden Sie unter [Networking für S3 Express One Zone](#).

Wenn Sie mit S3 Express One Zone arbeiten, sind API-Vorgänge auf Bucket-Ebene (Steuerebene) über einen regionalen Endpunkt verfügbar und werden als API-Vorgänge für regionale Endpunkte bezeichnet. Beispiele für API-Operationen für regionalen Endpunkte sind `CreateBucket` und `DeleteBucket`.

Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie zonale (Objektebene- oder Datenebenen-Endpunkt-API-Operationen) verwenden, um die Objekte in Ihrem Verzeichnis-Bucket hochzuladen und zu verwalten. API-Vorgänge für zonale Endpunkte sind über einen zonalen Endpunkt verfügbar. Beispiele für zonale API-Operationen sind `PutObject` und `CopyObject`.

Arbeiten mit S3 Express One Zone mithilfe der S3-Konsole AWS CLI, und AWS SDKs

Sie können mit der Speicherklasse und den Verzeichnis-Buckets von S3 Express One Zone arbeiten AWS SDKs, indem Sie die Amazon S3-Konsole AWS Command Line Interface (AWS CLI) und Amazon S3 REST API verwenden.

S3-Konsole

Um mit der Verwendung der S3-Konsole zu beginnen, führen Sie diese Schritte aus:

- [Erstellen eines Verzeichnis-Buckets](#)
- [Leeren eines Verzeichnis-Buckets](#)
- [Löschen eines Verzeichnis-Buckets](#)

Ein vollständiges Tutorial finden Sie unter [Tutorial: Erste Schritte mit S3 Express One Zone](#).

AWS SDKs

S3 Express One Zone unterstützt Folgendes AWS SDKs:

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java 2.x
- AWS SDK for JavaScript v3
- AWS SDK for .NET
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby
- AWS SDK for Kotlin
- AWS SDK for Rust

Wenn Sie mit S3 Express One Zone arbeiten, empfehlen wir die Verwendung der neuesten Version von AWS SDKs. Die AWS SDKs für S3 Express One Zone unterstützte Zone kümmert sich in Ihrem Namen um die Einrichtung, Aktualisierung und Beendigung von Sitzungen. Das bedeutet, dass Sie sofort mit dem API Betrieb beginnen können, nachdem Sie das heruntergeladen und installiert AWS SDKs und die erforderlichen IAM Berechtigungen konfiguriert haben. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

Informationen zu den AWS SDKs, einschließlich Informationen zum Herunterladen und Installieren, finden Sie unter [Tools, auf denen Sie aufbauen können AWS](#).

AWS SDK-Beispiele finden Sie nachfolgend.

- [Erstellen eines Verzeichnis-Buckets](#)
- [Leeren eines Verzeichnis-Buckets](#)
- [Löschen eines Verzeichnis-Buckets](#)

AWS Command Line Interface (AWS CLI)

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Verzeichnis-Buckets zu erstellen und unterstützte regionale und zonale API Endpunktoperationen für S3 Express One Zone zu verwenden.

Informationen zu den AWS CLI ersten Schritten finden [Sie unter Erste Schritte mit dem AWS CLI in der AWS CLI Befehlsreferenz](#).

Note

Um Directory-Buckets mit den [allgemeinen aws s3 Befehlen](#) zu verwenden, aktualisieren Sie Ihre Version AWS CLI auf die neueste Version. Weitere Informationen zur Installation und Konfiguration von finden Sie unter [Installieren oder Aktualisieren der neuesten Version von AWS CLI in der AWS CLI Befehlsreferenz](#). AWS CLI

AWS CLI Beispiele finden Sie in den folgenden Abschnitten:

- [Erstellen eines Verzeichnis-Buckets](#)
- [Leeren eines Verzeichnis-Buckets](#)
- [Löschen eines Verzeichnis-Buckets](#)

S3 Express API One-Zone-Betrieb

Die Speicherklasse Amazon S3 Express One Zone unterstützt sowohl regionale (Bucket-Ebene oder Kontrollebene) als auch zonale (Objektebene oder Datenebene) API Endpunktoperationen. Weitere Informationen erhalten Sie unter [Networking für S3 Express One Zone](#) und [Endpunkte und Gateway-Endpunkte VPC](#).

Regionale Endpunktoperationen API

Die folgenden regionalen API Endpunktoperationen werden für S3 Express One Zone unterstützt:

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [GetBucketPolicy](#)
- [ListDirectoryBuckets](#)
- [PutBucketPolicy](#)

Zonale Endpunktoperationen API

Die folgenden zonalen API Endpunktoperationen werden für S3 Express One Zone unterstützt:

- [CreateSession](#)
- [CopyObject](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [GetObject](#)
- [GetObjectAttributes](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListObjectsV2](#)
- [PutObject](#)
- [AbortMultipartUpload](#)
- [CompleteMultiPartUpload](#)
- [CreateMultipartUpload](#)
- [ListMultipartUploads](#)
- [ListParts](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [PutBucketEncryption](#)
- [GetBucketEncryption](#)

- [DeleteBucketEncryption](#)

Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten

Amazon S3 S3-Zugriffspunkte vereinfachen den Datenzugriff für alle AWS Service- oder Kundenanwendung, die Daten in S3 speichert. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Jeder Zugriffspunkt verfügt über unterschiedliche Berechtigungen und Netzwerkkontrollen, die S3 für alle Anforderungen anwendet, die über diesen Zugriffspunkt eingehen. Jeder Zugriffspunkt erzwingt eine benutzerdefinierte Zugriffspunktrichtlinie, die in Verbindung mit der Bucket-Richtlinie funktioniert, die dem zugrunde liegenden Bucket zugeordnet ist. Sie können jeden Access Point so konfigurieren, dass er nur Anfragen von einer virtuellen privaten Cloud (VPC) akzeptiert, um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk einzuschränken. Sie können auch benutzerdefinierte Block Public Access-Einstellungen für jeden Zugriffspunkt konfigurieren.

Note

- Sie können Zugriffspunkte nur zum Ausführen von Vorgängen an Objekten verwenden. Sie können mit Zugriffspunkten keine anderen Amazon S3-Vorgänge auszuführen, z. B. das Ändern oder Löschen von Buckets. Eine vollständige Liste der S3-Vorgänge, die Zugriffspunkte unterstützen, finden Sie unter [Kompatibilität des Zugangspunkts mit AWS service](#).
- Access Points funktionieren mit einigen, aber nicht mit allen AWS Dienste und Funktionen. Beispielsweise können Sie die regionsübergreifende Replikation nicht so konfigurieren, dass sie über einen Zugriffspunkt ausgeführt wird. Für eine vollständige Liste von AWS Dienste, die mit S3-Zugriffspunkten kompatibel sind, finden Sie unter [Kompatibilität des Zugangspunkts mit AWS service](#).

In diesem Abschnitt wird erläutert, wie Sie mit Amazon S3-Zugriffspunkten arbeiten. Hinweise zum Arbeiten mit Buckets finden Sie unter [Bucket-Übersicht](#). Weitere Informationen zur Arbeit mit Objekten finden Sie unter [Übersicht über Amazon-S3-Objekte](#).

Themen

- [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#)

- [Erstellen von Zugriffspunkten](#)
- [Verwenden von Zugriffspunkten](#)
- [Einschränkungen von Access Points](#)

Konfiguration von IAM Richtlinien für die Verwendung von Access Points

Unterstützung Amazon S3 S3-Zugriffspunkte AWS Identity and Access Management (IAM) Ressourcenrichtlinien, mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Damit eine Anwendung oder ein Benutzer über einen Zugriffspunkt auf Objekte zugreifen kann, müssen sowohl der Zugriffspunkt als auch der zugrunde liegende Bucket die Anforderung zulassen.

Important

Das Hinzufügen eines S3-Zugriffspunkts zu einem Bucket ändert das Verhalten des Buckets nicht, wenn auf den Bucket direkt über den Namen des Buckets oder den Amazon-Ressourcennamen (ARN) zugegriffen wird. Alle vorhandenen Vorgänge für den Bucket funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Zugriffspunktrichtlinie einschließen, gelten nur für Anforderungen, die über diesen Zugriffspunkt eingehen.

Wenn Sie IAM Ressourcenrichtlinien verwenden, achten Sie darauf, Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge von zu beheben AWS Identity and Access Management Access Analyzer bevor Sie Ihre Richtlinie speichern. IAM Access Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM [Richtliniengrammatik](#) und der [bewährten Methoden](#) zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten Empfehlungen, die Sie beim Erstellen von Richtlinien unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen.

Weitere Informationen zur Überprüfung von Richtlinien mithilfe von IAM Access Analyzer finden Sie unter [IAM Access Analyzer-Richtlinienuvalidierung](#) im IAM Benutzerhandbuch. Eine Liste der Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM [IAM Access Analyzer-Referenz zur Richtlinienüberprüfung](#).

Beispiele von -Zugriffspunktrichtlinien

Die folgenden Beispiele zeigen, wie IAM Richtlinien zur Steuerung von Anfragen über einen Access Point erstellt werden.

Note

Berechtigungen, die in einer Zugriffspunktrichtlinie erteilt werden, sind nur wirksam, wenn der zugrunde liegende Bucket auch denselben Zugriff zulässt. Sie können dies auf zwei Arten erreichen:

1. (Empfohlen) Delegieren Sie die Zugriffssteuerung vom Bucket an den Zugriffspunkt, wie unter [Delegieren der Zugangskontrolle an Zugriffspunkte](#) beschrieben.
2. Fügen Sie der Richtlinie des zugrunde liegenden Buckets dieselben Berechtigungen hinzu, die in der Zugriffspunktrichtlinie enthalten sind. Beispiel 1 für eine Zugriffspunktrichtlinie veranschaulicht, wie die zugrunde liegende Bucket-Richtlinie geändert wird, damit der erforderliche Zugriff gewährt wird.

Example 1 – Erteilung der Zugriffspunktberechtigung

Die folgende Zugriffspunktrichtlinie gewährt IAM Benutzern GET und *Jane* Benutzerkonten *123456789012* Berechtigungen für PUT Objekte mit dem Präfix *Jane/* über den Zugriffspunkt *my-access-point* im Konto *123456789012*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Jane"
      },
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/object/Jane/*"
    }
  ]
}
```

Note

Damit die Zugangspunkt-Richtlinie effektiv Zugriff gewährt *Jane*, muss der zugrundeliegende Bucket auch denselben Zugriff auf ermöglichen *Jane*. Sie können die Zugriffskontrolle vom Bucket an den Access Point delegieren, wie unter beschrieben [Delegieren der Zugangskontrolle an Zugriffspunkte](#). Oder Sie können dem zugrunde liegenden Bucket die folgende Richtlinie hinzufügen, um Jane die erforderlichen Berechtigungen zu erteilen. Beachten Sie, dass sich der Resource-Eintrag zwischen den Zugriffspunkt- und Bucket-Richtlinien unterscheidet.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Jane"
      },
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/Jane/*"
    }
  ]
}
```

Example 2 – Zugriffspunktrichtlinie mit Tag-Bedingung

Die folgende Zugriffspunktrichtlinie gewährt IAM dem Benutzer *Mateo* auf Rechnung *123456789012* Berechtigungen für GET Objekte über den Access Point *my-access-point* auf dem Konto *123456789012* die den Tag-Schlüssel haben *data* gesetzt mit einem Wert von *finanzieren*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Mateo"
      },
      "Action": "s3:GetObject",

```

```

    "Resource" : "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/
object/*",
    "Condition" : {
        "StringEquals": {
            "s3:ExistingObjectTag/data": "finance"
        }
    }
}
}]
}

```

Example 3 – Zugriffspunktrichtlinie, die eine Bucket-Auflistung ermöglicht

Die folgende Zugriffspunktrichtlinie erlaubt IAM Benutzern Arnav den Zugriff auf das Konto **123456789012** Berechtigung zum Anzeigen der Objekte, die sich in dem Bucket befinden, der dem Access Point zugrunde liegt **my-access-point** auf dem Konto **123456789012**.

```

{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Arnav"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point"
    }
  ]
}

```

Example 4 – Service-Kontrollrichtlinie

Gemäß der folgenden Dienststeuerungsrichtlinie müssen alle neuen Access Points mit einem Netzwerkursprung aus einer virtuellen privaten Cloud (VPC) erstellt werden. Mit dieser Richtlinie können Benutzer in Ihrer Organisation keine neuen Zugriffspunkte erstellen, auf die über das Internet zugegriffen werden kann.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",

```



```

    "Action": "s3:CreateAccessPoint",
    "Resource": "*",
    "Condition": {
      "StringNotEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
      }
    }
  }
}

```

Example 5 — Bucket-Richtlinie zur Beschränkung von S3-Vorgängen auf VPC Netzwerkquellen

Die folgende Bucket-Richtlinie beschränkt den Zugriff auf alle S3-Objektoperationen für den Bucket *amzn-s3-demo-bucket* auf Zugriffspunkte mit VPC Netzwerkursprung.

Important

Bevor Sie eine Anweisung wie in diesem Beispiel gezeigt verwenden, stellen Sie sicher, dass Sie keine Funktionen verwenden müssen, die von Zugriffspunkten nicht unterstützt werden, z. B. regionsübergreifende Replikation.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:BypassGovernanceRetention",
        "s3:DeleteObject",
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersion",
        "s3:DeleteObjectVersionTagging",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectLegalHold",
        "s3:GetObjectRetention",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",

```

```

        "s3:GetObjectVersionTagging",
        "s3:ListMultipartUploadParts",
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectLegalHold",
        "s3:PutObjectRetention",
        "s3:PutObjectTagging",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectVersionTagging",
        "s3:RestoreObject"
    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
        "StringNotEquals": {
            "s3:AccessPointNetworkOrigin": "VPC"
        }
    }
}
]
}
}

```

Bedingungsschlüssel

S3-Zugriffspunkte verfügen über Bedingungsschlüssel, die Sie in IAM Richtlinien verwenden können, um den Zugriff auf Ihre Ressourcen zu kontrollieren. Die folgenden Bedingungsschlüssel stellen nur einen Teil einer IAM Richtlinie dar. Vollständige Richtlinienbeispiele finden Sie unter [Beispiele von - Zugriffspunktrichtlinien](#), [the section called “Delegieren der Zugangskontrolle an Zugriffspunkte”](#) und [the section called “Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte”](#).

s3:DataAccessPointArn

Dieses Beispiel zeigt eine Zeichenfolge, die Sie für den Abgleich auf einem Access Point verwenden können. Das folgende Beispiel entspricht allen Access Points für AWS-Konto *123456789012* in der Region *us-west-2*:

```

"Condition" : {
  "StringLike": {
    "s3:DataAccessPointArn": "arn:aws:s3:us-west-2:123456789012:accesspoint/*"
  }
}

```

s3:DataAccessPointAccount

Dieses Beispiel zeigt einen Zeichenfolgenoperator, mit dem Sie die Konto-ID des Besitzers eines Zugriffspunkts abgleichen können. Das folgende Beispiel entspricht allen Access Points, die dem gehören AWS-Konto **123456789012**.

```
"Condition" : {
  "StringEquals": {
    "s3:DataAccessPointAccount": "123456789012"
  }
}
```

s3:AccessPointNetworkOrigin

Dieses Beispiel zeigt einen Zeichenfolgenoperator, den Sie verwenden können, um für den Netzwerkursprung entweder Internet oder VPC abzugleichen. Das folgende Beispiel entspricht nur Access Points mit einem VPC Ursprung.

```
"Condition" : {
  "StringEquals": {
    "s3:AccessPointNetworkOrigin": "VPC"
  }
}
```

Weitere Informationen zur Verwendung von Bedingungsschlüsseln mit Amazon S3 finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Delegieren der Zugangskontrolle an Zugriffspunkte

Sie können die Zugriffssteuerung für einen Bucket an die Zugriffspunkte des Buckets delegieren. Die folgende Bucket-Beispielrichtlinie ermöglicht Vollzugriff auf alle Zugriffspunkte, die dem Konto des Bucket-Eigentümers gehören. Somit wird der gesamte Zugriff auf diesen Bucket durch die an seine Zugriffspunkte angehängten Richtlinien gesteuert. Wir empfehlen, Ihre Buckets auf diese Weise für alle Anwendungsfälle zu konfigurieren, die keinen direkten Zugriff auf den Bucket erfordern.

Example 6 – Bucket-Richtlinie zum Delegieren der Zugriffskontrolle an Standardzugriffspunkte

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Principal" : { "AWS": "*" },
      "Action" : "*",
      "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
      "Condition": {
        "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
ID" }
      }
    }
  ]
}
```

Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte

Wenn Sie einen Zugriffspunkt für einen Bucket erstellen möchten, der einem anderen Konto gehört, müssen Sie zuerst den Zugriffspunkt erstellen, indem Sie den Bucket-Namen und die Kontobesitzer-ID angeben. Dann muss der Bucket-Eigentümer die Bucket-Richtlinie aktualisieren, um Anfragen vom Zugriffspunkt zu autorisieren. Das Erstellen eines Access Points ähnelt dem Erstellen eines Access DNS CNAME Points insofern, als der Access Point keinen Zugriff auf den Inhalt des Buckets gewährt. Der gesamte Bucket-Zugriff wird durch die Bucket-Richtlinie kontrolliert. Die folgende Beispiel-Bucket-Richtlinie erlaubt GET und LIST fordert den Bucket von einem Access Point aus an, der einem vertrauenswürdigen Unternehmen gehört AWS-Konto.

Ersetzen *Bucket ARN* mit dem ARN des Buckets.

Example 7 — Bucket-Richtlinie, mit der Berechtigungen an einen anderen delegiert werden AWS-Konto

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Principal" : { "AWS": "*" },
      "Action" : ["s3:GetObject","s3:ListBucket"],
      "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
    }
  ]
}
```

```
    "Condition": {
      "StringEquals" : { "s3:DataAccessPointAccount" : "Access point owner's
account ID" }
    }
  ]
}
```

Erstellen von Zugriffspunkten

Amazon S3 stellt Funktionen zum Erstellen und Verwalten von Zugriffspunkten bereit. Sie können S3-Zugriffspunkte erstellen, indem Sie AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDKs oder Amazon S3 REST API.

Standardmäßig können Sie bis zu 10.000 Access Points pro Region für jede Ihrer AWS-Konten. Wenn Sie mehr als 10.000 Access Points für ein einzelnes Konto in einer einzelnen Region benötigen, können Sie eine Erhöhung der Servicequote beantragen. Weitere Informationen zu Servicekontingenten und zur Beantragung einer Erhöhung finden Sie unter [AWS Service Quotas](#) in der Allgemeinen AWS-Referenz.

Note

Da Sie möglicherweise Ihren Zugriffspunktnamen veröffentlichen möchten, damit andere Benutzer den Zugriffspunkt verwenden können, sollten Sie vertrauliche Informationen im Namen des Zugriffspunkts vermeiden. Zugriffspunktnamen werden in einer öffentlich zugänglichen Datenbank veröffentlicht, die als Domain Name System (DNS) bezeichnet wird.

Regeln zur Benennung von Amazon S3-Zugriffspunkten

Ein Zugriffspunktname muss die folgenden Bedingungen erfüllen:

- Muss innerhalb einer einzigen Datei eindeutig sein AWS-Konto und Region
- Muss die DNS Namensbeschränkungen einhalten
- Muss mit einer Zahl oder einem Kleinbuchstaben beginnen
- Muss zwischen 3 und 50 Zeichen lang sein.
- Dürfen nicht mit einem Bindestrich (-) beginnen oder enden
- Dürfen keine Unterstriche (_), Großbuchstaben oder Punkte (.) enthalten

- Kann nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#).

Informationen zum Erstellen eines Zugriffspunkts finden Sie in den folgenden Themen.

Themen

- [Erstellen eines Zugriffspunkts](#)
- [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#)
- [Verwalten des öffentlichen Zugriffs auf Zugriffspunkte](#)

Erstellen eines Zugriffspunkts

Ein Zugriffspunkt ist genau einem Amazon S3-Bucket zugeordnet. Wenn Sie einen Bucket in Ihrem verwenden möchten AWS-Konto, müssen Sie zuerst einen Bucket erstellen. Weitere Informationen zum Erstellen von Buckets finden Sie unter [Erstellen, Konfigurieren und Arbeiten mit Amazon S3-Buckets](#).

Sie können auch einen kontoübergreifenden Access Point erstellen, der mit einem Bucket in einem anderen verknüpft ist AWS-Konto, sofern Sie den Bucket-Namen und die Konto-ID des Bucket-Besitzers kennen. Wenn Sie kontoübergreifende Zugriffspunkte erstellen, erhalten Sie jedoch erst Zugriff auf Daten im Bucket, wenn Ihnen vom Bucket-Eigentümer die entsprechenden Berechtigungen erteilt wurden. Der Bucket-Eigentümer muss dem Konto des Zugriffspunktbesitzers (Ihr Konto) über die Bucket-Richtlinie Zugriff auf den Bucket gewähren. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte](#).

Standardmäßig können Sie bis zu 10.000 Access Points pro Region für jede Ihrer AWS-Konten. Wenn Sie mehr als 10.000 Access Points für ein einzelnes Konto in einer einzelnen Region benötigen, können Sie eine Erhöhung der Servicequote beantragen. Weitere Informationen zu Servicekontingenten und zur Beantragung einer Erhöhung finden Sie unter [AWS Service Quotas](#) in der Allgemeine AWS-Referenz.

Die folgenden Beispiele zeigen, wie Sie einen Access Point mit dem erstellen AWS CLI und die S3-Konsole. Weitere Informationen zum Erstellen von Access Points mit dem REST API finden Sie unter [CreateAccessPoint](#) in der Amazon Simple Storage Service API Reference.

Verwenden der S3-Konsole

So erstellen Sie einen Zugriffspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Access Point erstellen möchten.
3. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
4. Wählen Sie auf der Seite Zugriffspunkte die Option Zugriffspunkt erstellen aus.
5. Geben Sie im Feld Name des Zugriffspunkts den gewünschten Namen für den Zugriffspunkt ein. Weitere Benennungen zur Benennung von Zugangspunkten finden Sie unter [Regeln zur Benennung von Amazon S3-Zugriffspunkten](#).
6. Geben Sie unter Bucket-Name den S3-Bucket an, den Sie mit dem Zugriffspunkt verwenden möchten.

Um in einen Bucket in Ihrem Konto zu verwenden, wählen Sie Bucket in diesem Konto auswählen aus und geben Sie den Namen des Buckets ein oder suchen Sie danach.

Um einen Bucket in einem anderen zu verwenden AWS-Konto, wählen Sie Spezifizieren Sie einen Bucket in einem anderen Konto und geben Sie AWS-Konto ID und Name des Buckets.

Note

Wenn Sie einen Bucket in einem anderen verwenden AWS-Konto, muss der Bucket-Besitzer die Bucket-Richtlinie aktualisieren, um Anfragen vom Access Point zu autorisieren. Ein Beispiel für eine Bucket-Richtlinie finden Sie in Beispiel [Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte](#).

7. Wählen Sie einen Ausgangspunkt des Netzwerks aus. Wenn Sie Virtual Private Cloud (VPC) wählen, geben Sie die VPCID ein, die Sie mit dem Access Point verwenden möchten.

Weitere Informationen zu Netzwerkursprüngen für Zugangspunkte finden Sie unter [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#).

8. Wählen Sie unter Block Public Access settings for this Access Point (Block-Public-Access-Einstellungen für diesen Zugriffspunkt) die Block-Public-Access-Einstellungen, die Sie für den

Zugriffspunkt anwenden möchten. Alle Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig für neue Zugriffspunkte aktiviert. Wir empfehlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine bestimmte Einstellung deaktivieren müssen.

Note

Nachdem Sie einen Zugriffspunkt erstellt haben, können Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs nicht mehr ändern.

Weitere Informationen zur Verwendung von Amazon S3-Block-Public-Access mit Zugriffspunkten finden Sie unter [Verwalten des öffentlichen Zugriffs auf Zugriffspunkte](#).

9. (Optional) Geben Sie unter Access point policy (Zugriffspunktrichtlinie) – optional die Zugriffspunktrichtlinie an. Beheben Sie vor dem Speichern Ihrer Richtlinie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge. Weitere Informationen zum Festlegen einer Zugriffspunktrichtlinie finden Sie unter [Beispiele von -Zugriffspunktrichtlinien](#).
10. Wählen Sie Create access point (Zugriffspunkt erstellen) aus.

Verwendung der AWS CLI

Der folgende Beispielbefehl erstellt einen Access Point mit dem Namen *example-ap* für den Eimer *amzn-s3-demo-bucket* auf dem Konto *111122223333*. Um den Access Point zu erstellen, senden Sie eine Anfrage an Amazon S3, die Folgendes spezifiziert:

- Den Namen des Zugriffspunkts Informationen zu Benennungsregeln finden Sie unter [the section called “Regeln zur Benennung von Amazon S3-Zugriffspunkten”](#).
- Den Name des Buckets, dem Sie den Zugriffspunkt zuweisen möchten
- Die Konto-ID für AWS-Konto dem der Bucket gehört.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --  
bucket amzn-s3-demo-bucket
```

Wenn Sie einen Access Point mithilfe eines Buckets in einem anderen erstellen AWS-Konto, fügen Sie den `--bucket-account-id` Parameter ein. Der folgende Beispielbefehl erstellt einen Access Point in AWS-Konto *111122223333*, unter Verwendung des Buckets *amzn-s3-demo-bucket2*, das ist in der AWS-Konto *444455556666*.


```
aws s3control create-access-point --name example-ap --account-id 111122223333 --  
bucket amzn-s3-demo-bucket --bucket-account-id 444455556666
```

Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind

Wenn Sie einen Access Point erstellen, können Sie wählen, ob Sie den Access Point über das Internet zugänglich machen möchten, oder Sie können angeben, dass alle Anfragen, die über diesen Access Point gestellt werden, von einer bestimmten virtuellen privaten Cloud stammen müssen (VPC). Ein Zugangspunkt, der über das Internet zugänglich ist, soll einen Netzwerkursprung von Internet haben. Er kann von überall im Internet verwendet werden, vorbehaltlich anderer Zugriffsbeschränkungen für den Zugriffspunkt, den zugrunde liegenden Bucket und die zugehörige Ressourcen, z. B. die angeforderten Objekte. Ein Access Point, auf den nur von einem bestimmten Access Point aus zugegriffen werden kann VPC, VPC hat einen Netzwerkursprung von, und Amazon S3 lehnt alle Anfragen an den Access Point ab, die nicht von diesem VPC stammen.

Important

Sie können den Netzwerkursprung eines Zugriffspunkts nur angeben, wenn Sie den Zugriffspunkt erstellen. Nachdem Sie den Zugriffspunkt erstellt haben, können Sie seinen Netzwerkursprung nicht mehr ändern.

Um einen Access Point auf den VPC reinen Zugriff zu beschränken, fügen Sie den `VpcConfiguration` Parameter der Anfrage zur Erstellung des Access Points bei. Im `VpcConfiguration` Parameter geben Sie die VPC ID an, mit der Sie den Access Point verwenden möchten. Wenn eine Anfrage über den Access Point gestellt wird, muss die Anfrage von dem stammen, VPC andernfalls lehnt Amazon S3 sie ab.

Sie können den Netzwerkursprung eines Access Points abrufen, indem Sie AWS CLI, AWS SDKs, oder REST APIs. Wenn für einen Access Point eine VPC Konfiguration angegeben ist, ist sein Netzwerkursprung VPC. Andernfalls ist der Netzwerkursprung des Zugriffspunkts Internet.

Example

Beispiel: Erstellen Sie einen Access Point, der auf den VPC Zugriff beschränkt ist

Im folgenden Beispiel wird ein Access Point mit dem Namen `example-vpc-ap` Bucket `example-bucket` in Account erstellt `123456789012`, der den Zugriff nur über den ermöglicht

vpc-1a2b3cVPC. Das Beispiel überprüft dann, ob der neue Zugriffspunkt den Netzwerkursprung VPC hat.

AWS CLI

```
aws s3control create-access-point --name example-vpc-ap --account-id 123456789012 --bucket example-bucket --vpc-configuration VpcId=vpc-1a2b3c
```

```
aws s3control get-access-point --name example-vpc-ap --account-id 123456789012

{
  "Name": "example-vpc-ap",
  "Bucket": "example-bucket",
  "NetworkOrigin": "VPC",
  "VpcConfiguration": {
    "VpcId": "vpc-1a2b3c"
  },
  "PublicAccessBlockConfiguration": {
    "BlockPublicAcls": true,
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true
  },
  "CreationDate": "2019-11-27T00:00:00Z"
}
```

Um einen Access Point mit a zu verwendenVPC, müssen Sie die Zugriffsrichtlinie für Ihren VPC Endpunkt ändern. VPCEndpunkte ermöglichen den Datenfluss von Ihnen VPC zu Amazon S3. Sie verfügen über Zugriffskontrollrichtlinien, die VPC kontrollieren, wie Ressourcen innerhalb von mit Amazon S3 interagieren dürfen. Anfragen von Ihnen VPC an Amazon S3 werden nur dann über einen Access Point erfolgreich ausgeführt, wenn die VPC Endpunktrichtlinie Zugriff sowohl auf den Access Point als auch auf den zugrunde liegenden Bucket gewährt.

Note

Um den Zugriff auf Ressourcen nur innerhalb eines zu ermöglichenVPC, stellen Sie sicher, dass Sie eine [private, gehostete Zone](#) für Ihren VPC Endpunkt einrichten. Um eine private gehostete Zone zu verwenden, [ändern Sie Ihre VPC Einstellungen](#) so, dass die

[VPCNetzwerkattribute](#) `enableDnsHostnames` und auf eingestellt `enableDnsSupport` sind `true`.

In der folgenden Beispiel-Richtlinienanweisung wird ein VPC Endpunkt so konfiguriert, dass Aufrufe `GetObject` für einen Bucket mit dem Namen `awsexamplebucket1` und einen Access Point mit dem Namen `example-vpc-ap` möglich sind.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::awsexamplebucket1/*",
        "arn:aws:s3:us-west-2:123456789012:accesspoint/example-vpc-ap/object/*"
      ]
    }
  ]
}
```

Note

Die "Resource" Deklaration in diesem Beispiel verwendet einen Amazon-Ressourcennamen (ARN), um den Access Point zu spezifizieren. Weitere Informationen zum Access Point ARNs finden Sie unter [Verwenden von Zugriffspunkten](#).

Weitere Informationen zu VPC Endpunktrichtlinien finden Sie [unter Verwenden von Endpunktrichtlinien für Amazon S3](#) im VPCBenutzerhandbuch.

Verwalten des öffentlichen Zugriffs auf Zugriffspunkte

Amazon S3-Zugriffspunkte unterstützen unabhängige Block Public Access-Einstellungen für jeden Zugriffspunkt. Wenn Sie einen Zugriffspunkt erstellen, können Sie die Block Public Access-Einstellungen für diesen Zugriffspunkt festlegen. Für jede Anforderung, die über einen Zugriffspunkt eingeht, wertet Amazon S3 die Block Public Access-Einstellungen für diesen Zugriffspunkt, den

zugrunde liegenden Bucket und das Konto des Bucket-Eigentümers aus. Wenn eine dieser Einstellungen darauf hinweist, dass die Anforderung gesperrt werden soll, lehnt Amazon S3 die Anforderung ab.

Weitere Hinweise zur Funktion S3 Block Public Access finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Important

- Alle Block Public Access-Einstellungen sind standardmäßig für Zugriffspunkte aktiviert. Sie müssen alle Einstellungen, die Sie nicht auf einen Zugriffspunkt anwenden möchten, explizit deaktivieren.
- Amazon S3 unterstützt derzeit nicht das Ändern der Public Block Access-Einstellungen eines Zugriffspunkts, nachdem der Zugriffspunkt erstellt wurde.

Example

Beispiel: Erstellen eines Zugriffspunkts mit benutzerdefinierten Block-Public-Access-Einstellungen

In diesem Beispiel wird ein Zugriffspunkt mit dem Namen `example-ap` für Bucket `example-bucket` in Konto `123456789012` mit nicht standardmäßigen Block Public Access-Einstellungen erstellt. Das Beispiel ruft dann die Konfiguration des neuen Zugriffspunkts ab, um seine Block Public Access-Einstellungen zu überprüfen.

AWS CLI

```
aws s3control create-access-point --name example-ap --account-id
123456789012 --bucket example-bucket --public-access-block-configuration
BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=true,RestrictPublicBuckets=t
```

```
aws s3control get-access-point --name example-ap --account-id 123456789012

{
  "Name": "example-ap",
  "Bucket": "example-bucket",
  "NetworkOrigin": "Internet",
  "PublicAccessBlockConfiguration": {
    "BlockPublicAcls": false,
    "IgnorePublicAcls": false,
```

```
    "BlockPublicPolicy": true,  
    "RestrictPublicBuckets": true  
  },  
  "CreationDate": "2019-11-27T00:00:00Z"  
}
```

Verwenden von Zugriffspunkten

Sie können auf die Objekte in einem Amazon S3 S3-Bucket mit einem Access Point zugreifen, indem Sie die Amazon S3 S3-Konsole verwenden. AWS CLI, AWS SDKs, oder der Amazon S3 RESTAPI.

Access Points haben Amazon-Ressourcennamen (ARNs). Access ARNs Points ähneln BucketARNs, sind aber explizit typisiert und kodieren die Region des Access Points und die AWS-Konto ID des Besitzers des Access Points. Weitere Informationen zu finden Sie ARNs unter [Amazon Resource Names \(ARNs\)](#) in der Allgemeine AWS-Referenz.

Access Point ARNs verwendet das Formatarn:aws:s3:region:account-id:accesspoint/resource. Beispielsweise:

- arn:aws:s3:us-west-2:123456789012:accesspoint/teststeht für den benannten Zugangspunkttest, der dem Konto gehört 123456789012 in der Region us-west-2.
- arn:aws:s3:us-west-2:123456789012:accesspoint/*steht für alle Access Points unter dem Konto 123456789012 in der Region us-west-2.

ARNsVerwenden Sie für Objekte, auf die über einen Access Point zugegriffen wird, das Formatarn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource. Beispielsweise:

- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01steht für das Objekt Einheit-01, auf den über den genannten Zugangspunkt zugegriffen wird Test, im Besitz eines Accounts 123456789012 in der Region us-west-2.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/*steht für alle Objekte für den genannten Access Point Test, im Konto 123456789012 in der Region us-west-2.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01/finance/*steht für alle Objekte unter dem Präfix Einheit-01/Finanzen/ für den genannten Access Point Test, im Konto 123456789012 in der Region us-west-2.

Zugreifen auf einen Bucket über einen S3-Zugriffspunkt

S3-Zugangspunkte unterstützen nur die virtual-host-style Adressierung. Um einen Bucket über einen Zugriffspunkt zu adressieren, verwenden Sie das folgende Format.

```
https://access-point-name-account-id.s3-accesspoint.region.amazonaws.com
```

Note

- Wenn der Name Ihres Access Points Bindestriche (-) enthält, schließen Sie die Bindestriche in URL und fügen Sie vor der Konto-ID einen weiteren Bindestrich ein. Um beispielsweise einen Access Point mit dem Namen zu verwenden *Finanzdokumente* gehört einem Konto *123456789012* in der Region *us-west-2*, das URL Passende wäre `https://finance-docs-123456789012.s3-accesspoint.us-west-2.amazonaws.com`.
- S3-Zugangspunkte unterstützen keinen Zugriff überHTTP. Access Points unterstützen nur den sicheren Zugriff überHTTPS.

Themen

- [Überwachen und Protokollieren von Zugriffspunkten](#)
- [Nutzen von Amazon S3-Zugriffspunkten mit der Amazon S3-Konsole](#)
- [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#)
- [Verwenden von Zugriffspunkten mit kompatiblen Amazon S3-Vorgänge](#)

Wenn Sie über eine Virtual Private Cloud (VPC) verfügen, finden Sie weitere Informationen unter [Verwaltung des Amazon S3 S3-Zugriffs mit VPC Endpunkten und S3-Zugriffspunkten](#).

Überwachen und Protokollieren von Zugriffspunkten

Amazon S3 protokolliert Anfragen, die über Access Points gestellt werden, und Anfragen an die API Operationen, die Access Points verwalten, gestellt werden, wie `CreateAccessPoint` und `GetAccessPointPolicy`. Um Nutzungsmuster zu überwachen und zu verwalten, können Sie auch Amazon CloudWatch Logs-Anforderungsmetriken für Access Points konfigurieren.

Themen

- [CloudWatch Metriken anfordern](#)
- [Anforderungsprotokolle](#)

CloudWatch Metriken anfordern

Um die Leistung von Anwendungen, die Access Points verwenden, zu verstehen und zu verbessern, können Sie Amazon S3 S3-Anforderungsmetriken verwenden CloudWatch . Anforderungsmetriken helfen Ihnen beim Überwachen von Amazon S3-Anforderungen, um Probleme bei der Ausführung schnell zu identifizieren und zu beheben.

Standardmäßig stehen diese Anforderungsmetriken auf Ebene der Buckets zur Verfügung. Sie können jedoch einen Filter für Anforderungsmetriken mit einem gemeinsamen Präfix, Objekt-Tags oder einem Zugriffspunkt definieren. Wenn Sie einen Zugriffspunktfilter erstellen, enthält die Konfiguration der Anforderungsmetriken Anforderungen an den von Ihnen angegebenen Zugriffspunkt. Sie können Metriken erhalten, Alarme festlegen und auf Dashboards zugreifen, um Echtzeit-Vorgänge anzuzeigen, die über diesen Zugriffspunkt ausgeführt werden.

Sie müssen sich dafür entscheiden, Metriken anzufordern, indem Sie sie in der Konsole konfigurieren oder Amazon S3 verwendenAPI. Die Anforderungsmetriken sind in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung verfügbar. Anforderungsmetriken werden zum gleichen Tarif wie CloudWatch benutzerdefinierte Metriken in Rechnung gestellt. Weitere Informationen finden Sie unter [CloudWatch Amazon-Preise](#).

Informationen zum Erstellen einer Anforderungsmetrikkonfiguration, die nach Zugriffspunkt filtert, finden Sie unter [Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert](#).

Anforderungsprotokolle

Sie können Anfragen protokollieren, die über Access Points gestellt wurden, und Anfragen an dieAPIs, die Access Points verwalten, gestellt wurden, z. `CreateAccessPoint` B. `GetAccessPointPolicy`, mithilfe der Serverzugriffsprotokollierung und AWS CloudTrail.

CloudTrail Protokolleinträge für Anfragen, die über Access Points gestellt wurden, enthalten den Access Point ARN im `resources` Abschnitt des Protokolls.

Angenommen, folgende Konfiguration liegt vor:

- Ein Bucket mit dem Namen *amzn-s3-demo-bucket1* in der Region *us-west-2* das enthält ein Objekt mit dem Namen *my-image.jpg*
- Ein Zugangspunkt mit dem Namen *my-bucket-ap* das ist verbunden mit *amzn-s3-demo-bucket1*
- Importieren in S3; AWS-Konto ID von *123456789012*

Das folgende Beispiel zeigt den `resources` Abschnitt eines CloudTrail Protokolleintrags für die vorherige Konfiguration:

```
"resources": [  
  {"type": "AWS::S3::Object",  
    "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1/my-image.jpg"},  
  },  
  {"accountId": "123456789012",  
    "type": "AWS::S3::Bucket",  
    "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1"},  
  },  
  {"accountId": "123456789012",  
    "type": "AWS::S3::AccessPoint",  
    "ARN": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-bucket-ap"},  
  }  
]
```

Weitere Informationen zu S3 Server-Zugriffsprotokollen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#). Weitere Informationen zur AWS CloudTrail, siehe [Was ist AWS CloudTrail?](#) in der AWS CloudTrail Benutzerleitfaden.

Nutzen von Amazon S3-Zugriffspunkten mit der Amazon S3-Konsole

In diesem Abschnitt wird erklärt, wie Sie Ihre Amazon S3 S3-Zugriffspunkte verwalten und verwenden, indem Sie AWS Management Console. Bevor Sie beginnen, navigieren Sie zur Detailseite für den Access Point, den Sie verwalten oder verwenden möchten, wie im folgenden Verfahren beschrieben.

Themen

- [Auflisten von Zugriffspunkten für Ihr Konto](#)
- [Auflisten von Zugriffspunkten für einen Bucket](#)
- [Anzeigen der Konfigurationsdetails für einen Zugriffspunkt](#)

- [Verwenden eines Zugriffspunkts](#)
- [Anzeigen der Block Public Access-Einstellungen für einen Zugriffspunkt](#)
- [Bearbeiten einer Zugriffspunktrichtlinie](#)
- [Erstellen eines Zugriffspunkts](#)

Auflisten von Zugriffspunkten für Ihr Konto

Um alle Access Points aufzulisten, die in Ihrem AWS-Konto

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, für die Sie Access Points auflisten möchten.
3. Klicken Sie im Navigationsbereich links in der Konsole auf Access points (Zugriffspunkte).
4. Sehen Sie sich auf der Seite Access Points unter Access Points die Access Points in Ihrem AWS-Region.
5. (Optional) Suchen Sie nach Zugriffspunkten nach Name, indem Sie einen Namen in das Textfeld neben dem Dropdown-Menü „Region“ eingeben.
6. Wählen Sie den Namen des Zugriffspunkts, den Sie verwalten oder verwenden möchten.

Auflisten von Zugriffspunkten für einen Bucket

Um alle Access Points in Ihnen aufzulisten AWS-Konto für einen einzelnen Eimer

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region Wählen Sie dann die Region aus, für die Sie Access Points auflisten möchten.
3. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Buckets.
4. Wählen Sie auf der Seite Buckets den Namen des Buckets aus, dessen Zugriffspunkte Sie auflisten möchten.
5. Wählen Sie auf der Bucket-Detailseite den Tab Access points (Zugriffspunkte).
6. Wählen Sie den Namen des Zugriffspunkts, den Sie verwalten oder verwenden möchten.

Anzeigen der Konfigurationsdetails für einen Zugriffspunkt

1. Navigieren Sie zur Detailseite des Zugriffspunkts für den Zugriffspunkt, dessen Details Sie anzeigen möchten, wie unter [Auflisten von Zugriffspunkten für Ihr Konto](#) beschrieben.
2. Zeigen Sie unter Access point overview (Zugriffspunkt-Übersicht) die Konfigurationsdetails und Eigenschaften für den ausgewählten Zugriffspunkt an.

Verwenden eines Zugriffspunkts


1. Navigieren Sie zur Detailseite des Zugriffspunkts für den gewünschten Zugriffspunkt, wie unter [Auflisten von Zugriffspunkten für Ihr Konto](#) beschrieben.
2. Wählen Sie im Tab Objects (Objekte) den Namen eines Objekts oder von Objekten aus, auf das/ die Sie über den Zugriffspunkt zugreifen möchten. Die Konsole zeigt über dem Namen Ihres Buckets eine Beschriftung mit dem aktuell von Ihnen verwendeten Zugriffspunkt an. Während Sie den Zugriffspunkt verwenden, können Sie nur die Objekt-Vorgänge ausführen, die durch die Zugriffspunktberechtigungen gewährt werden.

Note

- In der Konsolenansicht werden immer alle Objekte im Bucket angezeigt. Die Verwendung eines Zugriffspunkts, wie in dieser Prozedur beschrieben, beschränkt die Vorgänge, die Sie für diese Objekte ausführen können, jedoch nicht, ob Sie sehen können, dass sie im Bucket vorhanden sind.
- Die S3-Managementkonsole unterstützt nicht die Verwendung von Virtual Private Cloud (VPC) -Zugriffspunkten für den Zugriff auf Bucket-Ressourcen. Um von einem Access Point aus auf Bucket-Ressourcen VPC zuzugreifen, verwenden Sie den AWS CLI, AWS SDKs oder Amazon S3 REST APIs.

Anzeigen der Block Public Access-Einstellungen für einen Zugriffspunkt

1. Navigieren Sie zur Detailseite des Zugriffspunkts für den Zugriffspunkt, dessen Einstellungen Sie anzeigen möchten, wie unter [Auflisten von Zugriffspunkten für Ihr Konto](#) beschrieben.
2. Wählen Sie Permissions (Berechtigungen).
3. Überprüfen Sie unter Access point policy (Zugriffspunktrichtlinie) die Block-Public-Access-Einstellungen für den Zugriffspunkt.

 Note

Sie können die Block Public Access-Einstellungen für einen Zugriffspunkt nicht ändern, nachdem der Zugriffspunkt erstellt wurde.

Bearbeiten einer Zugriffspunktrichtlinie

1. Navigieren Sie zur Detailseite des Zugriffspunkts für den Zugriffspunkt, dessen Richtlinie Sie bearbeiten möchten, wie unter [Auflisten von Zugriffspunkten für Ihr Konto](#) beschrieben.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Access point policy (Zugriffspunktrichtlinie) Edit (Bearbeiten) aus.
4. Geben Sie die Richtlinie für den Zugriffspunkt in das Textfeld ein. Die Konsole zeigt automatisch den Amazon-Ressourcennamen (ARN) für den Access Point an, den Sie in der Richtlinie verwenden können.

Erstellen eines Zugriffspunkts

1. Navigieren Sie zur Liste der Zugriffspunkte für Ihr Konto oder für einen bestimmten Bucket, wie unter [Auflisten von Zugriffspunkten für Ihr Konto](#) beschrieben.
2. Aktivieren Sie die Optionsschaltfläche neben dem Namen des Zugriffspunkts, den Sie löschen möchten.
3. Wählen Sie Delete (Löschen).
4. Bestätigen Sie, dass Sie Ihren Zugriffspunkt löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und Delete (Löschen) wählen.

Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt

Wenn Sie einen Zugriffspunkt erstellen, generiert Amazon S3 automatisch einen Alias, den Sie anstelle eines Amazon S3-Bucket-Namens für den Datenzugriff verwenden können. Sie können diesen Access Point-Alias anstelle eines Amazon-Ressourcennamens (ARN) für den Betrieb auf der Datenebene des Access Points verwenden. Eine Liste dieser Vorgänge finden Sie unter [Kompatibilität des Zugangspunkts mit AWS service](#).

Im Folgenden finden Sie ein Beispiel ARN und einen Zugriffspunkt-Alias für einen Access Point mit dem Namen *my-access-point*.

- ARN – `arn:aws:s3:region:account-id:accesspoint/my-access-point`
- Zugriffspunkt-Alias – `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias`

Weitere Informationen zu finden Sie ARNs unter [Amazon Resource Names \(ARNs\)](#) in der Allgemeine AWS-Referenz.

Aliasname für Zugriffspunkte

Ein Zugriffspunkt-Aliasname wird innerhalb desselben Namespace wie ein Amazon S3-Bucket erstellt. Dieser Aliasname wird automatisch generiert und kann nicht geändert werden. Ein Zugriffspunkt-Aliasname erfüllt alle Anforderungen eines gültigen Amazon S3-Bucket-Namens und besteht aus den folgenden Teilen:

access point prefix-metadata-s3alias

Note

Das Suffix `-s3alias` ist für Zugriffspunkt-Aliasnamen reserviert und kann nicht für Bucket- oder Zugriffspunkt-Namen verwendet werden. Weitere Informationen zu Amazon-S3-Bucket-Benennungsregeln finden Sie unter [Regeln für die Benennung von Buckets](#).

Zugriffspunkt-Alias-Anwendungsfälle und -beschränkungen

Bei der Übernahme von Zugriffspunkten können Sie Zugriffspunkt-Aliasnamen verwenden, ohne dass umfangreiche Codeänderungen erforderlich sind.

Wenn Sie einen Zugriffspunkt erstellen, generiert Amazon S3 automatisch einen Zugriffspunkt-Aliasnamen, wie im folgenden Beispiel gezeigt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-access-point --bucket amzn-s3-demo-bucket1 --name my-access-point
--account-id 111122223333
{
```

```

"AccessPointArn":
  "arn:aws:s3:region:111122223333:accesspoint/my-access-point",
"Alias": "my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-s3alias"
}

```

Sie können diesen Zugriffspunkt-Aliasnamen anstelle eines Amazon S3-Bucket-Namen für jeden Datenebenen-Vorgang verwenden. Eine Liste dieser Vorgänge finden Sie unter [Kompatibilität des Zugriffspunkts mit AWS service](#).

Folgendes AWS CLI Das Beispiel für den `get-object` Befehl verwendet den Access Point-Alias des Buckets, um Informationen über das angegebene Objekt zurückzugeben. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```

aws s3api get-object --bucket my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-
s3alias --key dir/my_data.rtf my_data.rtf

{
  "AcceptRanges": "bytes",
  "LastModified": "2020-01-08T22:16:28+00:00",
  "ContentLength": 910,
  "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
  "VersionId": "null",
  "ContentType": "text/rtf",
  "Metadata": {}
}

```

Einschränkungen

- Aliase können nicht von Kunden konfiguriert werden.
- Aliasse können auf einem Zugriffspunkt nicht gelöscht, geändert oder deaktiviert werden.
- Sie können diesen Zugriffspunkt-Aliasnamen anstelle eines Amazon-S3-Bucket-Namen für manche Datenebenen-Operationen verwenden. Eine Liste dieser Vorgänge finden Sie unter [Zugriffspunkt-Kompatibilität mit S3-Vorgänge](#).
- Sie können einen Aliasnamen für Amazon S3-Kontrollebenenvorgänge nicht verwenden. Eine Liste der Vorgänge auf der Amazon S3-Steuerebene finden Sie unter [Amazon S3 Control](#) in der Amazon Simple Storage Service API Reference.
- Sie können S3-Zugriffspunkt-Aliase nicht als Quelle oder Ziel für Move-Operationen in der Amazon S3 S3-Konsole verwenden.
- Aliase können nicht verwendet werden in AWS Identity and Access Management (IAM) Richtlinien.

- Aliase können nicht als Protokollierungsziel für S3-Server-Zugriffsprotokolle verwendet werden.
- Aliase können nicht als Ziel für die Protokollierung verwendet werden für AWS CloudTrail Protokolle.
- Amazon unterstützt SageMaker GroundTruth keine Access Point-Aliase.

Verwenden von Zugriffspunkten mit kompatiblen Amazon S3-Vorgänge

Die folgenden Beispiele veranschaulichen, wie Zugriffspunkte mit kompatiblen Vorgängen in Amazon S3 verwendet werden.

Themen

- [Kompatibilität des Zugangspunkts mit AWS service](#)
- [Zugriffspunkt-Kompatibilität mit S3-Vorgänge](#)
- [Anfordern eines Objekts über einen Zugriffspunkt](#)
- [Hochladen eines Objekts über einen Zugriffspunkt-Alias](#)
- [Löschen eines Objekts über einen Zugriffspunkt](#)
- [Auflisten von Objekten über einen Zugriffspunkt-Alias](#)
- [Hinzufügen eines Tag-Satzes zu einem Objekt über einen Zugriffspunkt](#)
- [Erteilen Sie Zugriffsberechtigungen über einen Access Point mithilfe eines ACL](#)

Kompatibilität des Zugangspunkts mit AWS service

Mit Aliasnamen von Amazon-S3-Zugriffspunkten können Anwendungen, die einen S3-Bucketnamen benötigen, problemlos einen Zugriffspunkt verwenden. Sie können S3-Zugriffspunkt-Aliasnamen überall verwenden, wo Sie S3-Bucketnamen verwenden, um auf Daten in S3 zuzugreifen. Weitere Informationen finden Sie unter [Zugriffspunkt-Alias-Anwendungsfälle und -beschränkungen](#).

Zugriffspunkt-Kompatibilität mit S3-Vorgänge

Sie können Access Points verwenden, um mithilfe der folgenden Teilmenge von Amazon S3 APIs auf einen Bucket zuzugreifen. Alle unten aufgeführten Operationen können entweder Access Point ARNs - oder Access Point-Aliase akzeptieren:

S3-Operationen

- [AbortMultipartUpload](#)

- [CompleteMultipartUpload](#)
- [CopyObject](#)(Nur Kopien derselben Region)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [DeleteObjectTagging](#)
- [GetBucketAcl](#)
- [GetBucketCors](#)
- [GetBucketLocation](#)
- [GetBucketNotificationConfiguration](#)
- [GetBucketPolicy](#)
- [GetObject](#)
- [GetObjectAcl](#)
- [GetObjectAttributes](#)
- [GetObjectLegalHold](#)
- [GetObjectRetention](#)
- [GetObjectTagging](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjects](#)
- [ListObjectsV2](#)
- [ListObjectVersions](#)
- [ListParts](#)
- [Presign](#)
- [PutObject](#)
- [PutObjectLegalHold](#)
- [PutObjectRetention](#)
- [PutObjectAcl](#)

- [PutObjectTagging](#)
- [RestoreObject](#)
- [UploadPart](#)
- [UploadPartCopy](#)(Nur Kopien aus derselben Region)

Anfordern eines Objekts über einen Zugriffspunkt

Im folgenden Beispiel wird das Objekt angefordert *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*, und speichert die heruntergeladene Datei als *download.jpg*.

AWS CLI

```
aws s3api get-object --key my-image.jpg --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod download.jpg
```

Hochladen eines Objekts über einen Zugriffspunkt-Alias

Das folgende Beispiel lädt das Objekt hoch *my-image.jpg* über den Access Point-Alias, der der Konto-ID *my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b*-s3alias gehört *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api put-object --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias --key my-image.jpg --body my-image.jpg
```

Löschen eines Objekts über einen Zugriffspunkt

Im folgenden Beispiel wird das Objekt gelöscht *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api delete-object --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod --key my-image.jpg
```


Auflisten von Objekten über einen Zugriffspunkt-Alias

Im folgenden Beispiel werden Objekte anhand des Access Point-Alias aufgelistet, der der Konto-ID *my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b*-s3alias gehört *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api list-objects-v2 --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias
```

Hinzufügen eines Tag-Satzes zu einem Objekt über einen Zugriffspunkt

Im folgenden Beispiel wird dem vorhandenen Objekt ein Tag-Set hinzugefügt *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api put-object-tagging --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod --key my-image.jpg --tagging TagSet=[{Key="finance",Value="true"}]
```

Erteilen Sie Zugriffsberechtigungen über einen Access Point mithilfe eines ACL

Im folgenden Beispiel wird ein ACL auf ein vorhandenes Objekt angewendet *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api put-object-acl --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod --key my-image.jpg --acl private
```

Einschränkungen von Access Points

Es gelten die folgenden Einschränkungen und Beschränkungen für Amazon S3-Zugriffspunkte:

- Jeder Zugriffspunkt ist genau einem Bucket zugeordnet, den Sie beim Erstellen des Zugriffspunkts angeben müssen. Nachdem Sie einen Zugriffspunkt erstellt haben, können Sie ihn keinem anderen

Bucket zuordnen. Sie können jedoch einen Zugriffspunkt löschen und dann einen anderen mit demselben Namen erstellen. Diesen neuen Zugriffspunkt können Sie dann einem anderen Bucket zuordnen.

- Namen von Zugriffspunkten müssen bestimmte Bedingungen erfüllen. Weitere Informationen zur Benennung von Zugangspunkten finden Sie unter [Regeln zur Benennung von Amazon S3-Zugriffspunkten](#).
- Nachdem Sie einen Access Point erstellt haben, können Sie seine Virtual Private Cloud (VPC) - Konfiguration nicht mehr ändern.
- Zugriffspunkt-Richtlinien sind auf eine Größe von 20 KB beschränkt.
- Sie können maximal 10.000 Zugriffspunkte pro AWS-Konto pro Region. Wenn Sie mehr als 10 000 Zugriffspunkte für ein einzelnes Konto in einer Region benötigen, können Sie eine Erhöhung der Service Quotas beantragen. Weitere Informationen zu Servicekontingenten und zur Beantragung einer Erhöhung finden Sie unter [AWS Servicekontingenten](#) in der Allgemeine AWS-Referenz.
- In AWS-Regionen Wenn Sie über mehr als 1.000 Access Points verfügen, können Sie in der Amazon S3 S3-Konsole nicht anhand des Namens nach einem Access Point suchen.
- Sie können einen Zugriffspunkt nicht als Ziel für die Replikation in S3 verwenden. Weitere Informationen zur Replikation finden Sie unter [Überblick über das Replizieren von Objekten](#).
- Sie können S3-Zugriffspunkt-Aliase nicht als Quelle oder Ziel für Move-Operationen in der Amazon S3 S3-Konsole verwenden.
- Sie können Access Points nur adressieren, indem Sie virtual-host-style URLs. Weitere Hinweise virtual-host-style zur Adressierung finden Sie unter [Zugreifen auf einen Amazon-S3-Bucket und Auflisten des Buckets](#).
- APIVorgänge, die die Funktionalität des Access Points steuern (z. B. `PutAccessPoint` und `GetAccessPointPolicy`), unterstützen keine kontenübergreifenden Anrufe.
- Sie müssen verwenden AWS Signature Version 4, wenn Sie Anfragen an einen Access Point mit dem stellen REST APIs. Weitere Informationen zur Authentifizierung von Anfragen finden Sie unter [Authentifizieren von Anfragen \(AWS Signature Version 4\)](#) in der Amazon Simple Storage API Service-Referenz.
- Access Points unterstützen nur Anfragen über HTTPS. Amazon S3 antwortet automatisch mit einer HTTP Weiterleitung auf alle Anfragen HTTP, auf die Sie die Anfrage aktualisieren möchten HTTPS.
- Zugriffspunkte unterstützen keinen anonymen Zugriff.
- Mit den kontoübergreifenden Zugriffspunkten erhalten Sie jedoch erst dann Zugriff auf Daten, wenn Ihnen vom Bucket-Eigentümer die entsprechenden Berechtigungen erteilt wurden. Der

Bucket-Eigentümer behält immer die ultimative Kontrolle über den Datenzugriff und muss die Bucket-Richtlinie aktualisieren, um Anforderungen vom kontoübergreifenden Zugriffspunkt zu autorisieren. Eine Bucket-Beispielrichtlinie finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#).

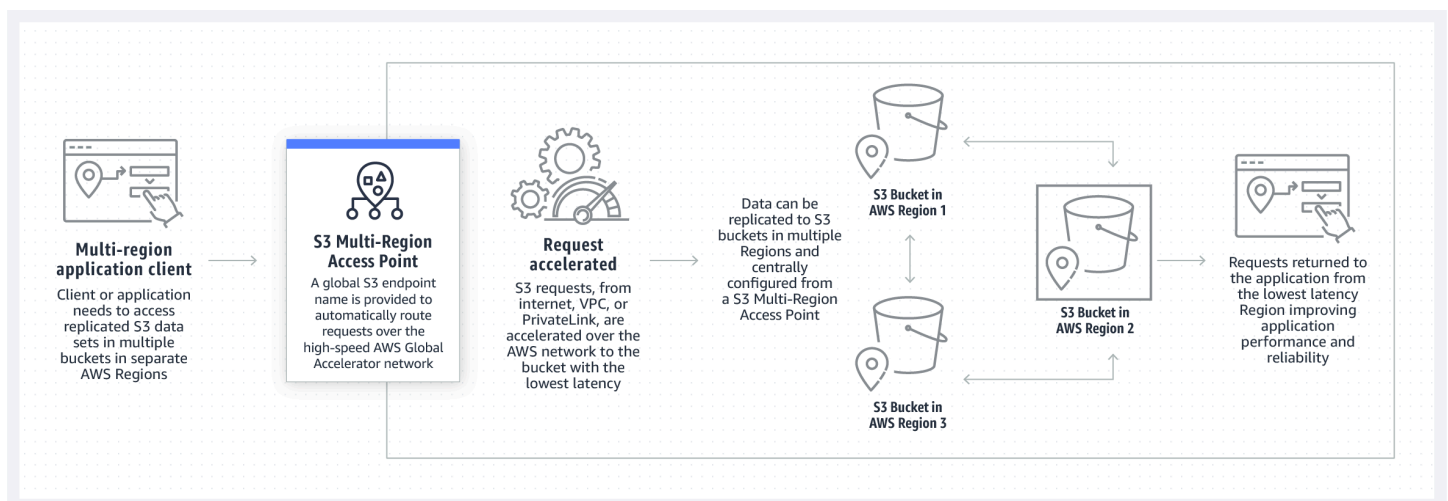
- Wenn Sie in der Amazon-S3-Konsole einen kontoübergreifenden Zugriffspunkt anzeigen, wird in der Spalte Zugriff Unbekannt angezeigt. Die Amazon-S3-Konsole kann nicht feststellen, ob öffentlicher Zugriff für den zugehörigen Bucket und die zugehörigen Objekte gewährt wurde. Sofern Sie keine öffentliche Konfiguration für einen bestimmten Anwendungsfall benötigen, empfehlen wir Ihnen und dem Bucket-Besitzer, den gesamten öffentlichen Zugriff auf den Access Point und den Bucket zu blockieren. Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Multi-Regions-Zugriffspunkte in Amazon S3

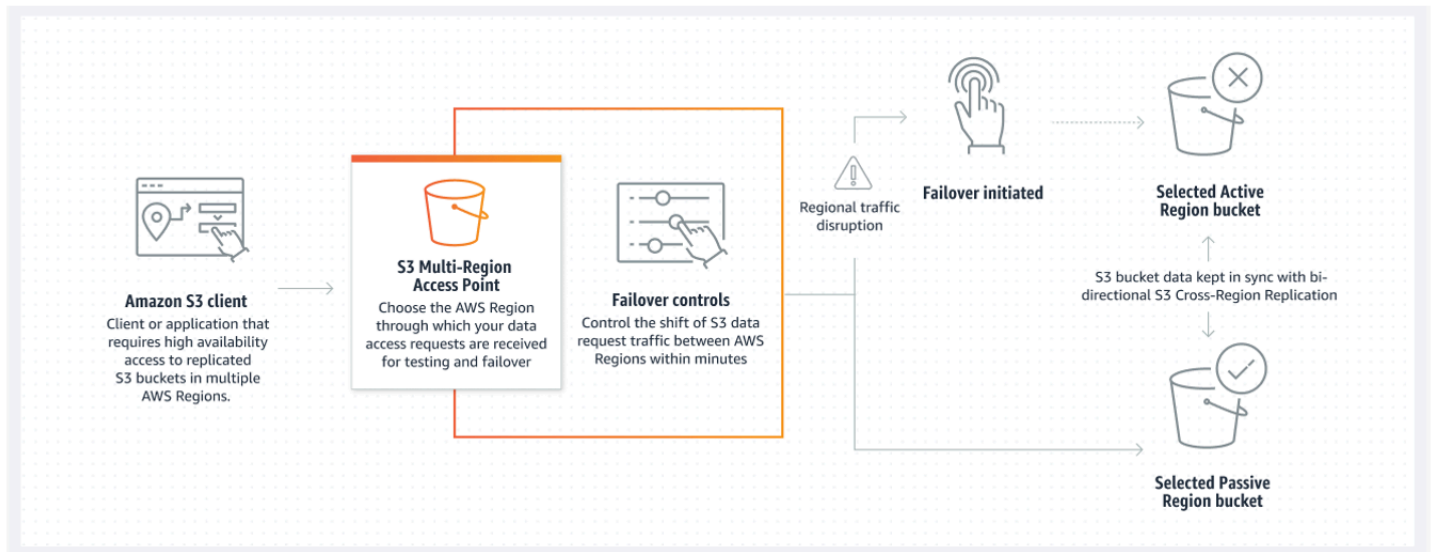
Amazon S3 Multi-Region Access Points stellen einen globalen Endpunkt bereit, mit dem Anwendungen Anforderungen von S3-Buckets ausführen können, die sich in mehreren AWS-Regionen befinden. Sie können Multi-Regions Access Points verwenden, um Multi-Regions-Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Anstatt Anforderungen über das überlastete öffentliche Internet zu senden, bieten Multi-Regions-Zugriffspunkte integrierte Netzwerkausfallsicherheit mit Beschleunigung internetbasierter Anforderungen an Amazon S3. Anwendungsanfragen, die an einen globalen Access Point-Endpunkt mit mehreren Regionen gestellt werden [AWS Global Accelerator](#), werden automatisch über das AWS globale Netzwerk an den nächstgelegenen S3-Bucket mit aktivem Routing-Status weitergeleitet.

Wenn Sie einen Multi-Region-Access Point erstellen, geben Sie an, AWS-Regionen wo Sie Daten speichern möchten, die über diesen Multi-Region-Access Point bereitgestellt werden sollen. Sie können [S3-regionsübergreifende Replikation \(CRR\)](#) verwenden, um Daten zwischen Buckets in diesen Regionen zu synchronisieren. Anschließend können Sie Daten über den globalen Multi-Regions-Zugriffspunkt-Endpunkt anfordern oder schreiben. Amazon S3 übermittelt Anforderungen automatisch an den replizierten Datensatz aus der nächstgelegenen verfügbaren Region. Multi-Region Access Points sind auch mit Anwendungen kompatibel, die in Amazon Virtual Private Clouds (VPCs) ausgeführt werden, einschließlich Anwendungen, die [AWS PrivateLink für Amazon S3](#) verwenden.

Die folgende Abbildung ist eine grafische Darstellung eines Amazon S3 Multi-Region Access Points in einer Aktiv-Aktiv-Konfiguration. Die Grafik zeigt, wie Amazon-S3-Anforderungen automatisch an Buckets in der nächstgelegenen aktiven AWS-Region weitergeleitet werden.



Die folgende Abbildung ist eine grafische Darstellung eines Amazon S3 Multi-Region Access Points in einer Aktiv-Passiv-Konfiguration. Die Grafik zeigt, wie Sie den Amazon-S3-Datenzugriffsverkehr so steuern können, dass ein Failover zwischen aktiven und passiven AWS-Regionen erfolgt.



Weitere Informationen zur Verwendung von Multi-Region Access Points finden Sie unter [Tutorial: Getting started with Amazon S3 Multi-Region Access Points](#).

Themen

- [Erstellen Multi-Regions-Zugriffspunkten](#)
- [Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink](#)
- [Stellen von Anforderungen über einen Multi-Region Access Point](#)

Erstellen Multi-Regions-Zugriffspunkten

Gehen Sie wie folgt vor, um einen Multi-Region Access Point in Amazon S3 zu erstellen:

- Geben Sie den Namen für den Multi-Region Access Point an.
- Wählen Sie in jedem Bucket AWS-Region, in dem Sie Anfragen für den Multi-Region Access Point bearbeiten möchten, aus.
- Konfigurieren Sie die Einstellungen für Amazon S3 Block Public Access für den Multi-Region Access Point.

Sie geben alle diese Informationen in einer Erstellungsanforderung an, die Amazon S3 asynchron verarbeitet. Amazon S3 bietet ein Token, mit dem Sie den Status der asynchronen Erstellungsanforderung überwachen können.

Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge von AWS Identity and Access Management Access Analyzer bevor Sie Ihre Richtlinie speichern. IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM [Richtliniengrammatik](#) und der [bewährten Methoden](#) zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten umsetzbare Empfehlungen, die Sie beim Erstellen von Richtlinien unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen. Weitere Informationen zur Überprüfung von Richtlinien mit IAM Access Analyzer finden Sie unter [IAMAccess Analyzer-Richtlinienvvalidierung](#) im IAMBenutzerhandbuch. Eine Liste der Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM [IAMAccess Analyzer-Referenz zur Richtlinienüberprüfung](#).

Wenn Sie den verwendenAPI, erfolgt die Anforderung zum Erstellen eines Access Points mit mehreren Regionen asynchron. Wenn Sie eine Anforderung zum Erstellen eines Multi-Regions-Zugriffspunkts übermitteln, autorisiert Amazon S3 die Anforderung synchron. Es gibt dann sofort ein Token zurück, mit dem Sie den Fortschritt der Erstellungsanforderung verfolgen können. Weitere Informationen zum Nachverfolgen asynchroner Anforderungen zum Erstellen und Verwalten von Multi-Regions-Zugriffspunkten finden Sie unter [Verwendung von Access Points mit mehreren Regionen mit unterstützten Vorgängen API](#).

Nachdem Sie den Multi-Regions-Zugriffspunkt erstellt haben, können Sie eine Zugriffssteuerungsrichtlinie für diesen erstellen. Jedem Multi-Regions-Zugriffspunkt kann eine Richtlinie zugeordnet sein. Eine Richtlinie für Multi-Region Access Points ist eine ressourcenbasierte Richtlinie, mit der Sie die Verwendung des Multi-Region Access Point nach Ressource, Benutzer oder anderen Bedingungen einschränken können.

Note

Damit eine Anwendung oder ein Benutzer über einen Multi-Regions Access Point auf ein Objekt zugreifen kann, müssen die folgenden Richtlinien beide die Anforderung zulassen:

- Die Zugriffsrichtlinie für den Multi-Region Access Point
- Die Zugriffsrichtlinie für den zugrunde liegenden Bucket, der das Objekt enthält

Wenn die beiden Richtlinien unterschiedlich sind, hat die restriktivere Richtlinie Vorrang.

Um die Verwaltung von Berechtigungen für Multi-Region Access Points zu vereinfachen, können Sie die Zugriffssteuerung vom Bucket an den Multi-Region Access Point delegieren. Weitere Informationen finden Sie unter [the section called “Beispielrichtlinien für Multi-Region Access Points”](#).

Die Verwendung eines Buckets mit einem Multi-Region-Access Point ändert nichts am Verhalten des Buckets, wenn auf den Bucket über den vorhandenen Bucket-Namen oder einen Amazon-Ressourcennamen (ARN) zugegriffen wird. Alle vorhandenen Vorgänge für den Bucket funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Multi-Regions-Zugriffspunktrichtlinie einschließen, gelten nur für Anforderungen, die über den Multi-Regions-Zugriffspunkt eingehen.

Sie können die Richtlinie für einen Multi-Regions-Zugriffspunkt nach dem Erstellen aktualisieren, die Richtlinie jedoch nicht löschen. Sie können jedoch die Richtlinie für Multi-Region Access Points aktualisieren, um alle Berechtigungen zu verweigern.

Themen

- [Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten](#)
- [Regeln für die Auswahl von Buckets für Amazon S3-Multi-Regions-Zugriffspunkte](#)
- [Erstellen eines Amazon S3-Multi-Region Access Point](#)
- [Blockieren des öffentlichen Zugriffs mit Amazon S3-Multi-Regions-Zugriffspunkten](#)
- [Anzeigen der Konfigurationsdetails für Amazon S3 Multi-Region Access Points](#)
- [Löschen eines Multi-Region Access Point](#)

Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten

Wenn Sie einen Multi-Regions-Zugriffspunkt erstellen, geben Sie ihm einen Namen, bei dem es sich um eine von Ihnen ausgewählte Zeichenfolge handelt. Sie können den Namen des Multi-Regions-Zugriffspunkts nach der Erstellung nicht mehr ändern. Der Name muss in Ihrem AWS-Konto einzigartig sein, und es muss den Namensanforderungen entsprechen, die in [Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts](#) aufgelistet sind. Wenn Sie den Multi-Regions-Zugriffspunkt identifizieren möchten, verwenden Sie einen Namen, der für Sie oder Ihre Organisation aussagekräftig ist oder der das Szenario widerspiegelt.

Sie verwenden diesen Namen beim Aufrufen von Sie verwenden diesen Namen, wenn Sie Verwaltungsvorgänge für Multi-Regions-Zugriffspunkten, z. B. `GetMultiRegionAccessPoint`

und `PutMultiRegionAccessPointPolicy`. Der Name wird nicht verwendet, um Anforderungen an den Multi-Region Access Point zu senden, und er muss Clients, die Anforderungen unter Verwendung des Multi-Region Access Point stellen, nicht offengelegt werden.

Wenn Amazon S3 einen Multi-Regions-Zugriffspunkt erstellt, weist er ihm automatisch einen Alias zu. Dieser Alias ist eine eindeutige alphanumerische Zeichenfolge, die mit `.mrp` endet. Der Alias wird verwendet, um den Hostnamen und den Amazon-Ressourcennamen (ARN) für einen Access Point mit mehreren Regionen zu erstellen. Der vollqualifizierte Name basiert auch auf dem Alias für den Multi-Regions-Zugriffspunkt.

Sie können den Namen eines Multi-Regions-Zugriffspunkts nicht anhand seines Alias ermitteln, sodass Sie einen Alias offenlegen können, ohne den Namen, den Zweck oder den Besitzer des Multi-Regions-Zugriffspunkts zu enthüllen. Amazon S3 wählt den Alias für jeden neuen Multi-Regions-Zugriffspunkt aus, und der Alias kann nicht geändert werden. Weitere Informationen zur Adressierung eines Multi-Regions-Zugriffspunkts finden Sie unter [Stellen von Anforderungen über einen Multi-Region Access Point](#).

Multi-Regions-Zugriffspunkt-Aliase sind im Laufe der Zeit eindeutig und basieren nicht auf dem Namen oder der Konfiguration eines Multi-Regions-Zugriffspunkts. Wenn Sie einen Multi-Regions-Zugriffspunkt erstellen, ihn dann löschen und einen anderen mit demselben Namen und derselben Konfiguration erstellen, hat der zweite Multi-Regions-Zugriffspunkt einen anderen Alias als der erste. Neue Multi-Regions-Zugriffspunkte können nie denselben Alias wie ein vorheriger Multi-Regions-Zugriffspunkt haben.

Regeln für die Auswahl von Buckets für Amazon S3-Multi-Regions-Zugriffspunkte

Jeder Multi-Regions-Zugriffspunkt ist den Regionen zugeordnet, in denen Sie Anforderungen erfüllen möchten. Der Multi-Regions-Zugriffspunkt muss genau einem Bucket in jeder dieser Regionen zugeordnet sein. Sie geben den Namen jedes Buckets in der Anforderung an, um den Multi-Regions-Zugriffspunkt zu erstellen. Buckets, die den Multi-Region-Access Point unterstützen, können sich entweder in demselben Bereich befinden AWS-Konto, dem der Multi-Region-Access Point gehört, oder sie können sich in einem anderen befinden. AWS-Konten

Ein einzelner Bucket kann von mehreren Multi-Regions-Zugriffspunkten verwendet werden.

⚠ Important

- Sie können die Buckets, die einem Multi-Regions-Zugriffspunkt zugeordnet sind, nur zum Zeitpunkt seiner Erstellung angeben. Nach der Erstellung können Sie keine Buckets hinzufügen, ändern oder aus der Konfiguration des Multi-Regions-Zugriffspunkts entfernen. Um die Buckets zu ändern, müssen Sie den gesamten Multi-Regions-Zugriffspunkt löschen und einen neuen erstellen.
- Sie können einen Bucket, der Teil eines Multi-Regions-Zugriffspunkts ist, nicht löschen. Wenn Sie einen Bucket löschen möchten, der einem Multi-Region Access Point zugeordnet ist, löschen Sie zuerst den Multi-Region Access Point.
- Wenn Sie Ihrem Multi-Region Access Point einen Bucket hinzufügen, dessen Eigentümer ein anderes Konto ist, muss der Bucket-Eigentümer ebenfalls seine Bucket-Richtlinie aktualisieren, um Zugriffsberechtigungen für den Multi-Region Access Point zu gewähren. Andernfalls kann der Multi-Region Access Point keine Daten aus diesem Bucket abrufen. Beispielrichtlinien, die zeigen, wie ein solcher Zugriff gewährt wird, finden Sie unter [Beispielrichtlinien für Multi-Region Access Points](#).
- Nicht alle Regionen unterstützen Multi-Regions-Zugriffspunkte. Eine Liste der unterstützten Regionen finden Sie unter [Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts](#).

Sie können Replikationsregeln erstellen, um Daten zwischen Buckets zu synchronisieren. Mit diesen Regeln können Sie Daten automatisch aus Quell-Buckets in Ziel-Buckets kopieren. Die Verbindung von Buckets mit einem Multi-Regions-Zugriffspunkt hat keinen Einfluss auf die Funktionsweise der Replikation. Das Konfigurieren der Replikation mit Multi-Regions-Zugriffspunkten wird in einem späteren Abschnitt beschrieben.

⚠ Important

Wenn Sie eine Anforderung an einen Multi-Region Access Point stellen, kennt der Multi-Region Access Point die Dateninhalte der Buckets in dem Multi-Region Access Point nicht. Daher kann es sein, dass der Bucket, der die Anforderung erhält, die angeforderten Daten nicht enthält. Um konsistente Datensätze in den Amazon S3 S3-Buckets zu erstellen, die einem Access Point mit mehreren Regionen zugeordnet sind, empfehlen wir, die

regionsübergreifende S3-Replikation () zu konfigurieren. CRR Weitere Informationen finden Sie unter [Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points](#).

Erstellen eines Amazon S3-Multi-Region Access Point

Das folgende Beispiel zeigt, wie ein Multi-Region Access Point über die Amazon-S3-Konsole erstellt werden kann.

Verwenden der S3-Konsole

So erstellen Sie einen Multi-Regions-Zugriffspunkt


1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie Multi-Region Access Points erstellen aus, um mit der Erstellung Ihres Multi-Region Access Point zu beginnen.
4. Geben Sie auf der Seite Multi-Region Access Point einen Namen für den Multi-Region Access Point in das Feld Name des Multi-Region Access Point ein.
5. Wählen Sie die Buckets aus, die diesem Multi-Region Access Point zugeordnet werden sollen. Sie können Buckets auswählen, die sich in Ihrem Konto befinden, oder Sie können Buckets aus anderen Konten auswählen.

Note

Sie müssen mindestens einen Bucket aus Ihrem Konto oder anderen Konten hinzufügen. Beachten Sie außerdem, dass Multi-Region-Access Points nur einen Bucket pro AWS-Region unterstützen. Daher können Sie nicht zwei Buckets aus derselben Region hinzufügen. [AWS-Regionen , die standardmäßig deaktiviert sind](#), werden nicht unterstützt.


- Um einen Bucket hinzuzufügen, der sich in Ihrem Konto befindet, wählen Sie Hinzufügen von Buckets aus. Es wird eine Liste aller Buckets in Ihrem Konto angezeigt. Sie können nach Ihrem Bucket anhand des Namens suchen oder die Bucket-Namen in alphabetischer Reihenfolge sortieren.

- Um einen Bucket aus einem anderen Konto hinzuzufügen, wählen Sie Bucket aus einem anderen Konto hinzufügen aus. Stellen Sie sicher, dass Sie den genauen Namen und die AWS-Konto ID des Buckets kennen, da Sie in anderen Konten nicht nach Buckets suchen oder danach suchen können.

 Note


Sie müssen eine gültige AWS-Konto ID und einen gültigen Bucket-Namen eingeben. Der Bucket muss sich auch in einer unterstützten Region befinden, andernfalls tritt ein Fehler auf, wenn Sie versuchen, Ihren Multi-Region Access Point zu erstellen. Eine Liste der Regionen, die Multi-Region Access Points unterstützen, finden Sie unter [Einschränkungen und Beschränkungen von Multi-Region Access Points](#).

6. (Optional) Wenn Sie einen hinzugefügten Bucket entfernen müssen, wählen Sie Entfernen aus.

 Note

Nach dem Erstellen des Multi-Region Access Point können Sie diesem keine Buckets hinzufügen und keine Buckets daraus entfernen.

7. Wählen Sie unter Block Public Access settings for this Multi-Region Access Point (Einstellungen für die Sperrung des öffentlichen Zugriffs für diesen Multi-Regions-Zugriffspunkt) die Einstellungen für die Sperrung des öffentlichen Zugriffs, die Sie für den Multi-Regions-Zugriffspunkt anwenden möchten. Alle Block-Einstellungen des öffentlichen Zugriffs sind standardmäßig für neue Multi-Regions-Zugriffspunkte aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine bestimmte Notwendigkeit haben, eine von ihnen zu deaktivieren.

 Note

Nach dem Erstellen eines Multi-Region Access Point können Sie die Block-Public-Access-Einstellungen für den Multi-Region Access Point nicht mehr ändern. Wenn Sie den öffentlichen Zugriff blockieren möchten, stellen Sie daher sicher, dass Ihre Anwendungen ohne öffentlichen Zugriff ordnungsgemäß funktionieren, bevor Sie einen Multi-Region Access Point erstellen.

8. Wählen Sie Multi-Regions-Zugriffspunkt erstellen.

⚠ Important

Wenn Sie Ihrem Multi-Region Access Point einen Bucket hinzufügen, dessen Eigentümer ein anderes Konto ist, muss der Bucket-Eigentümer auch seine Bucket-Richtlinie aktualisieren, um Zugriffsberechtigungen für den Multi-Region Access Point zu gewähren. Andernfalls kann der Multi-Region Access Point keine Daten aus diesem Bucket abrufen. Beispielrichtlinien, die zeigen, wie ein solcher Zugriff gewährt wird, finden Sie unter [Beispielrichtlinien für Multi-Region Access Points](#).

Verwenden Sie den AWS CLI

Sie können den verwenden AWS CLI , um einen Access Point für mehrere Regionen zu erstellen. Beim Erstellen des Multi-Region Access Point müssen Sie alle Buckets bereitstellen, die dieser unterstützen soll. Sie können dem Multi-Region Access Point nach der Erstellung keine Buckets hinzuzufügen.

Im folgenden Beispiel wird ein Multi-Region Access Point mit zwei Buckets über die AWS CLI erstellt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-multi-region-access-point --account-id 111122223333 --details '{
  "Name": "simple-multiregionaccesspoint-with-two-regions",
  "PublicAccessBlock": {
    "BlockPublicAcls": true,
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true
  },
  "Regions": [
    { "Bucket": "amzn-s3-demo-bucket1" },
    { "Bucket": "amzn-s3-demo-bucket2" }
  ]
}' --region us-west-2
```

Blockieren des öffentlichen Zugriffs mit Amazon S3-Multi-Regions-Zugriffspunkten

Jeder Multi-Regions-Zugriffspunkt hat verschiedene Block-Einstellungen des öffentlichen Zugriffs in Amazon S3. Diese Einstellungen funktionieren zusammen mit den Einstellungen für den Block Public

Access für denjenigen AWS-Konto, der den Multi-Region Access Point und die zugrunde liegenden Buckets besitzt.

Wenn Amazon S3 eine Anforderung autorisiert, wird die restriktivste Kombination dieser Einstellungen angewendet. Wenn die Block-Public-Access-Einstellungen für eine dieser Ressourcen (das Konto, das Eigentümer des Multi-Region Access Point ist, der zugrunde liegende Bucket oder das Bucket-Eigentümerkonto) den Zugriff für die angeforderte Aktion oder Ressource blockieren, lehnt Amazon S3 die Anforderung ab.

Alle Block-Einstellungen des öffentlichen Zugriffs sind standardmäßig für neue Zugriffspunkte aktiviert, und wir empfehlen Ihnen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie einen bestimmten Wert deaktivieren müssen. Alle Block-Einstellungen des öffentlichen Zugriffs sind standardmäßig für Multi-Regions-Zugriffspunkte aktiviert. Wenn Block Public Access aktiviert ist, kann der Multi-Region Access Point keine internetbasierten Anforderungen annehmen.

Important

Nach dem Erstellen eines Multi-Region Access Point können Sie die Block-Public-Access-Einstellungen für den Access Point nicht mehr ändern.

Weitere Informationen über Block Public Access in Amazon S3 finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Anzeigen der Konfigurationsdetails für Amazon S3 Multi-Region Access Points


Das folgende Beispiel zeigt, wie Konfigurationsdetails für einen Multi-Region Access Point über die Amazon-S3-Konsole angezeigt werden können.

Verwenden der S3-Konsole

So erstellen Sie einen Multi-Regions-Zugriffspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Point aus, dessen Konfigurationsdetails Sie anzeigen möchten.

- Auf der Registerkarte Eigenschaften werden alle Buckets aufgeführt, die mit Ihrem Access Point in mehreren Regionen verknüpft sind, das Erstellungsdatum, der Amazon-Ressourcenname (ARN) und der Alias. In der Spalte für die AWS-Konto-ID sind auch alle Buckets aufgeführt, deren Eigentümer externe Konten sind und die Ihrem Multi-Region-Access Point zugeordnet sind.
- Auf der Registerkarte Berechtigungen sind die Block-Public-Access-Einstellungen aufgeführt, die für die diesem Multi-Regions-Zugriffspunkt zugeordneten Buckets gelten. Sie können sich die Richtlinie für Multi-Region Access Points für Ihren Multi-Region Access Point auch ansehen, falls Sie eine erstellt haben. In der Info-Warnung auf der Seite Berechtigungen sind außerdem alle Buckets (in Ihrem Konto und anderen Konten) für diesen Multi-Region Access Point aufgeführt, für die die Einstellung Öffentlicher Zugriff ist blockiert aktiviert ist.
- Die Registerkarte Replikation und Failover bietet eine Kartenansicht der Buckets, die Ihrem Multi-Region Access Point zugeordnet sind, sowie der Regionen, in denen sich die Buckets befinden. Wenn es Buckets von einem anderen Konto gibt, für das Sie keine Berechtigung zum Abrufen von Daten haben, ist die Region in der Replikationsübersicht rot markiert, um darauf hinzuweisen, dass es sich um eine AWS-Region handelt, bei der Fehler beim Abrufen des Replikationsstatus aufgetreten sind.

 Note

Um Informationen zum Replikationsstatus aus einem Bucket in einem externen Konto abzurufen, muss Ihnen der Bucket-Eigentümer in seiner Bucket-Richtlinie die Berechtigung `s3:GetBucketReplication` erteilen.

Auf dieser Registerkarte sind auch die Replikationsmetriken, Replikationsregeln und Failover-Status für die Regionen aufgeführt, die mit Ihrem Multi-Region Access Point verwendet werden.

Mit dem AWS CLI

Sie können den verwenden AWS CLI , um die Konfigurationsdetails für einen Access Point mit mehreren Regionen anzuzeigen.

Im folgenden AWS CLI Beispiel wird Ihre aktuelle Multi-Region-Access Point-Konfiguration abgerufen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point --account-id 111122223333 --name amzn-s3-demo-bucket1
```

Löschen eines Multi-Region Access Point

Das folgende Verfahren zeigt, wie ein Multi-Region Access Point über die Amazon-S3-Konsole gelöscht werden kann.

Beim Löschen eines Multi-Region Access Point werden die dem Multi-Region Access Point zugeordneten Buckets nicht gelöscht, nur der Multi-Region Access Point selbst.

Verwenden der S3-Konsole

So löschen Sie einen Multi-Region Access Point

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie das Optionsfeld neben dem Namen Ihres Multi-Region Access Point aus.
4. Wählen Sie Löschen aus.
5. Geben Sie im Dialogfeld „Multi-Region Access Point löschen“ den Namen des AWS Buckets ein, den Sie löschen möchten.

Note

Stellen Sie sicher, dass Sie einen gültigen Bucket-Namen eingeben. Andernfalls wird die Schaltfläche Löschen deaktiviert.

6. Wählen Sie Löschen aus, um zu bestätigen, dass Sie Ihren Multi-Region Access Point tatsächlich löschen möchten.

Verwenden Sie den AWS CLI

Sie können den verwenden AWS CLI , um einen Access Point mit mehreren Regionen zu löschen. Durch diese Aktion werden die dem Multi-Region Access Point zugeordneten Buckets nicht gelöscht, nur der Multi-Region Access Point selbst. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-multi-region-access-point --account-id 123456789012 --details
Name=example-multi-region-access-point-name
```

Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink

Sie können Multi-Region Access Points verwenden, um Amazon-S3-Anforderungsdatenverkehr zwischen AWS-Regionen weiterzuleiten. Jeder globale Endpunkt von Multi-Region Access Points leitet den Amazon-S3-Datenanforderungsverkehr aus mehreren Quellen weiter, ohne dass Sie komplexe Netzwerkkonfigurationen mit separaten Endpunkten erstellen müssen. Zu diesen Quellen des Datenanforderungsverkehrs gehören:

- Datenverkehr aus einer Virtual Private Cloud (VPC)
- Datenverkehr aus On-Premises-Rechenzentren, der über AWS PrivateLink läuft
- Datenverkehr aus dem öffentlichen Internet

Wenn Sie eine AWS PrivateLink-Verbindung zu einem S3 Multi-Region Access Point herstellen, können Sie S3-Anforderungen in AWS oder zwischen mehreren AWS-Regionen über eine private Verbindung weiterleiten, indem Sie eine einfache Netzwerkarchitektur und -konfiguration verwenden. Wenn Sie AWS PrivateLink verwenden, müssen Sie keine VPC-Peering-Verbindung konfigurieren.

Themen

- [Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink](#)
- [Entfernen des Zugriffs auf einen Multi-Region Access Point von einem VPC-Endpunkt](#)

Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink

AWS PrivateLink bietet Ihnen private Konnektivität zu Amazon S3 mithilfe von privaten IP-Adressen in Ihrer virtuellen privaten Cloud (VPC). Sie können einen oder mehrere Schnittstellenendpunkte in Ihrem Gerät bereitstellen, um eine Verbindung VPC zu Amazon S3 Multi-Region Access Points herzustellen.

Sie können `com.amazonaws.s3-global.accesspoint`-Endpunkte für Access Points mit mehreren Regionen über, oder erstellen. AWS Management Console AWS CLI AWS SDKs [Weitere Informationen zur Konfiguration eines Schnittstellenendpunkts für Multi-Region-Access Points finden Sie unter Schnittstellenendpunkte im Benutzerhandbuch. VPC VPC](#)

Gehen Sie wie folgt vor, um über Schnittstellenendpunkte Anfragen an einen Multi-Region-Access Point zu richten, um den VPC und den Multi-Region-Access Point zu konfigurieren.

So konfigurieren Sie einen Access Point mit mehreren Regionen zur Verwendung mit AWS PrivateLink

1. Erstellen Sie einen geeigneten VPC Endpunkt, der eine Verbindung zu Access Points mit mehreren Regionen herstellen kann, oder verfügen Sie über einen entsprechenden Endpunkt. Weitere Informationen zum Erstellen von VPC Endpunkten finden Sie unter [VPCSchnittstellenendpunkte](#) im VPC Benutzerhandbuch.

Important

Stellen Sie sicher, dass Sie einen `com.amazonaws.s3-global.accesspoint`-Endpunkt erstellen. Andere Endpunkttypen können nicht auf Multi-RegionsZugriffspunkte zugreifen.

Nachdem dieser VPC Endpunkt erstellt wurde, werden alle Multi-Region-Access Point-Anfragen auf dem Endpunkt über diesen Endpunkt VPC weitergeleitet, sofern Sie für den Endpunkt private Zugriffe DNS aktiviert haben. Dies ist standardmäßig aktiviert.

2. Wenn die Richtlinie für multiregionale Access Points keine Verbindungen von VPC Endpunkten unterstützt, müssen Sie sie aktualisieren.
3. Stellen Sie sicher, dass die einzelnen Bucket-Richtlinien den Zugriff auf die Benutzer des Multi-Regions-Zugriffspunkts ermöglichen.

Denken Sie daran, dass Multi-Regions-Zugriffspunkte funktionieren, indem sie Anforderungen an Buckets weiterleiten, nicht indem sie selbst Anforderungen erfüllen. Dies ist wichtig, da der Absender der Anforderung über Berechtigungen für den Multi-Regions-Zugriffspunkt verfügen muss und auf die einzelnen Buckets im Multi-Regions-Zugriffspunkt zugreifen kann. Andernfalls wird die Anforderung möglicherweise an einen Bucket weitergeleitet, in dem der Originator keine Berechtigungen hat, um die Anforderung zu erfüllen. Ein Access Point mit mehreren Regionen und die zugehörigen Buckets können demselben oder einem anderen Konto gehören. AWS Ein Multi-Region-Access Point kann jedoch VPCs von unterschiedlichen Konten aus verwendet werden, sofern die Berechtigungen korrekt konfiguriert sind.

Aus diesem Grund muss die VPC Endpunktrichtlinie den Zugriff sowohl auf den Multi-Region-Zugriffspunkt als auch auf alle zugrunde liegenden Buckets ermöglichen, die Sie in der Lage sein sollen, Anfragen zu bearbeiten. Angenommen, Sie haben einen Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`. Es wird von Buckets unterstützt `amzn-s3-demo-bucket1` und `amzn-s3-demo-bucket2` alle gehören einem AWS Account. `123456789012` In diesem Fall würde die folgende VPC Endpunktrichtlinie es ermöglichen, dass `GetObject` Anfragen von den VPC gestellten Benutzern von einem der beiden Backing-Buckets erfüllt werden. `mfzwi23gnjvgw.mrap`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Read-buckets-and-MRAP-VPCE-policy",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1/*",
        "arn:aws:s3:::amzn-s3-demo-bucket2/*",
        "arn:aws:s3:::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
      ]
    }
  ]
}
```

Wie bereits erwähnt, müssen Sie außerdem sicherstellen, dass die Richtlinie für Access Points für mehrere Regionen so konfiguriert ist, dass sie den Zugriff über einen VPC Endpunkt unterstützt. Sie müssen den VPC Endpunkt, der den Zugriff anfordert, nicht angeben. Die folgende Beispielrichtlinie

würde jedem Anforderer Zugriff gewähren, der versucht, den Multi-Region Access Point für die GetObject-Anforderungen zu nutzen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Open-read-MRAP-policy",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
    }
  ]
}
```

Und natürlich benötigen die einzelnen Buckets jeweils eine Richtlinie zur Unterstützung des Zugriffs auf Anfragen, die über den VPC Endpunkt eingereicht wurden. Die folgende Beispielrichtlinie gewährt allen anonymen Benutzern Lesezugriff, was auch Anfragen einschließt, die über den VPC Endpunkt gestellt wurden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Public-read",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket2/*"
      ]
    }
  ]
}
```

Weitere Informationen zum Bearbeiten einer VPC Endpunktrichtlinie finden Sie unter [Steuern des Zugriffs auf Dienste mit VPC Endpunkten](#) im VPCBenutzerhandbuch.

Entfernen des Zugriffs auf einen Multi-Region Access Point von einem VPC-Endpunkt

Wenn Sie einen Multi-Region Access Point besitzen und den Zugriff darauf von einem Schnittstellenendpunkt entfernen möchten, müssen Sie eine neue Zugriffsrichtlinie für den Multi-Region Access Point bereitstellen, die den Zugriff für Anforderungen über VPC-Endpunkte verhindert. Wenn die Buckets in Ihrem Multi-Region Access Point Anforderungen über VPC-Endpunkte unterstützen, werden sie diese Anforderungen allerdings weiterhin unterstützen. Wenn Sie diesen Support verhindern möchten, müssen Sie auch die Richtlinien für die Buckets aktualisieren. Durch die Bereitstellung einer neuen Zugriffsrichtlinie für den Multi-Region Access Point wird nur der Zugriff auf den Multi-Region Access Point verhindert, nicht auf die zugrunde liegenden Buckets.

Note

Sie können eine Zugriffsrichtlinie für einen Multi-Regions-Zugriffspunkt nicht löschen. Um den Zugriff auf einen Multi-Regions-Zugriffspunkt zu entfernen, müssen Sie eine neue Zugriffsrichtlinie mit dem gewünschten geänderten Zugriff bereitstellen.

Anstatt die Zugriffsrichtlinie für den Multi-Region Access Point zu aktualisieren, können Sie die Bucket-Richtlinien aktualisieren, um Anforderungen über VPC-Endpunkte zu verhindern. In diesem Fall können Benutzer weiterhin über den VPC-Endpunkt auf den Multi-Region Access Point zugreifen. Wenn der Multi-Region Access Point jedoch an einen Bucket weitergeleitet wird, in dem die Bucket-Richtlinie den Zugriff verhindert, wird eine Fehlermeldung generiert.

Stellen von Anforderungen über einen Multi-Region Access Point

Wie andere Ressourcen verfügen auch Amazon S3 Multi-Region Access Points über Amazon-Ressourcennamen (ARNs). Sie können diese ARNs verwenden, um Anforderungen über die AWS Command Line Interface (AWS CLI), AWS-SDKs oder die Amazon-S3-API an Multi-Region Access Points zu leiten. Sie können diese ARNs auch verwenden, um Multi-Region Access Points in Zugriffskontrollrichtlinien zu identifizieren. Der ARN eines Multi-Region Access Point beinhaltet den Namen des Multi-Region Access Points nicht und gibt diesen nicht an. Weitere Informationen zur Verwendung von ARNs finden Sie unter [Amazon-Ressourcennamen \(ARN\)](#) im Allgemeine AWS-Referenz.

 Note

Der Alias des Multi-Region Access Point und der ARN können nicht gegeneinander ausgetauscht werden.

ARNs von Multi-Region Access Points verwenden das folgende Format:

```
arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias
```

Im Folgenden finden Sie einige Beispiele für ARNs von Multi-Region Access Points:

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap` repräsentiert den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, der im Besitz von AWS-Konto 123456789012 ist.
- `arn:aws:s3::123456789012:accesspoint/*` repräsentiert alle Multi-Region Access Points im Konto 123456789012. Dieser ARN stimmt mit allen Multi-Region Access Points für das Konto 123456789012 überein, entspricht aber keinen regionalen Amazon S3 Access Points, da der ARN keine AWS-Region enthält. Im Gegensatz dazu stimmt der ARN `arn:aws:s3:us-west-2:123456789012:accesspoint/*` mit allen regionalen Amazon S3 Access Points in der Region `us-west-2` für das Konto 123456789012 überein, entspricht aber keinem Multi-Region Access Point.

ARNs für Objekte, auf die über einen Multi-Region Access Point zugegriffen wird, verwenden folgendes Format:

```
arn:aws:s3::account_id:accesspoint/MultiRegionAccessPoint_alias//key
```

Wie bei ARNs von Multi-Region Access Points enthalten die ARNs für Objekte, auf die über Multi-Region Access Points zugegriffen wird, keine AWS-Region. Hier sind einige Beispiele.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01` repräsentiert `-01`, auf das über den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, im Besitz von Konto 123456789012, zugegriffen wird.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//*` repräsentiert alle Objekte, auf die über den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, in Konto 123456789012, zugegriffen werden kann.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01/finance/*` repräsentiert alle Objekte, auf die unter `-01/finance/` für den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, in Konto `123456789012`, zugegriffen werden kann.

Hostnamen für Multi-Regions-Zugriffspunkte

Sie können über einen Multi-Region Access Point auf Daten in Amazon S3 zugreifen, indem Sie den Hostnamen des Multi-Region Access Points verwenden. Anforderungen können aus dem öffentlichen Internet an diesen Hostnamen gerichtet werden. Wenn Sie ein oder mehrere Internet-Gateways für den Multi-Region Access Point konfiguriert haben, können Anforderungen auch aus einer Virtual Private Cloud (VPC) an diesen Hostnamen weitergeleitet werden. Weitere Informationen zum Erstellen von VPC-Schnittstellenendpunkten zur Verwendung mit mehreren Multi-Regions-Zugriffspunkten finden Sie unter [Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink](#).

Wenn Sie Anforderungen über einen Multi-Region Access Point von einer VPC unter Verwendung eines VPC-Endpunkts stellen möchten, können Sie AWS PrivateLink verwenden. Wenn Sie Anforderungen an einen Multi-Region Access Point mithilfe von AWS PrivateLink stellen, können Sie den Namen eines endpunktspezifischen regionalen DOMAIN NAME SYSTEM (DNS), der mit `region.vpce.amazonaws.com` endet, nicht direkt verwenden. Diesem Hostnamen wird kein Zertifikat zugeordnet, so dass er nicht direkt verwendet werden kann. Sie können den DOMAIN NAME SYSTEM (DNS)-Namen des VPC-Endpunkts weiterhin als CNAME- oder ALIAS-Ziel verwenden. Alternativ können Sie privates DOMAIN NAME SYSTEM (DNS) auf dem Endpunkt aktivieren und den standardmäßigen DNS-Namen `MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com` des Multi-Region Access Point wie in diesem Abschnitt beschrieben verwenden.

Wenn Sie Anforderungen an die API für Amazon-S3-Datenoperationen (z. B. `GetObject`) über einen Multi-Region Access Point stellen, lautet der Hostname für die Anforderung wie folgt:

`MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com`

Wenn Sie zum Beispiel eine `GetObject`-Anforderung über den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap` stellen möchten, stellen Sie eine Anforderung an den Hostnamen `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Der `s3-global`-Teil des Hostnamens gibt an, dass dieser Hostname nicht für eine bestimmte Region bestimmt ist.

Das Senden von Anforderungen über einen Multi-Regions-Zugriffspunkt ähnelt der Ausführung von Anforderungen über einen Zugriffspunkt für einzelne Regionen. Es ist jedoch wichtig, folgende Unterschiede zu beachten:

- ARNs von Multi-Region Access Points enthalten keine AWS-Region. Sie folgen dem Format `arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`.
- Für Anforderungen, die über API-Operationen gestellt werden (diese Anforderungen erfordern keinen ARN), verwenden Multi-Region Access Points ein anderes Endpunktschema. Das Schema ist `MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com` – z. B. `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Beachten Sie die Unterschiede im Vergleich zu einem Zugriffspunkt für einzelne Regionen:
 - Hostnamen des Multi-Regions-Zugriffspunkts verwenden ihren Alias, nicht den Namen des Multi-Regions-Zugriffspunkts.
 - Hostnamen von Multi-Region Access Points enthalten nicht die AWS-Konto-ID des Besitzers.
 - Hostnamen von Multi-Region Access Points enthalten keine AWS-Region.
 - Zu den Multi-Regions-Hostnamen gehören `s3-global.amazonaws.com` anstelle von `s3.amazonaws.com`.
- Anforderungen für Multi-Region Access Points müssen mit Signature Version 4A (SigV4A) signiert werden. Wenn Sie die AWS-SDKs verwenden, konvertiert das SDK ein SigV4 automatisch in SigV4A. Stellen Sie daher sicher, dass Ihr [AWS-SDK](#) SigV4A als Signaturimplementierung unterstützt, die zum Signieren der globalen Anforderungen der AWS-Region verwendet wird. Weitere Informationen zu SigV4A finden Sie unter [Signieren von AWS-API-Anforderungen](#) in der Allgemeine AWS-Referenz.

Multi-Regions-Zugriffspunkte und Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration ist eine Funktion, die schnelle Datenübertragungen zu Buckets ermöglicht. Transfer Acceleration wird auf der Ebene des einzelnen Buckets konfiguriert. Weitere Informationen zu Transfer Acceleration finden Sie unter [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#).

Multi-Region Access Points verwenden einen ähnlichen beschleunigten Übertragungsmechanismus wie Transfer Acceleration, um große Objekte über das AWS-Netzwerk zu senden. Aus diesem Grund müssen Sie Transfer Acceleration nicht verwenden, wenn Sie Anforderungen über einen Multi-Region Access Point senden. Diese erhöhte Übertragungsleistung wird automatisch in den Multi-Region Access Point integriert.

Themen

- [Berechtigungen](#)
- [Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts](#)
- [Weiterleitung der Multi-Regions-Zugriffspunktanforderung](#)
- [Failover-Kontrollen für Amazon S3 Multi-Region Access Points](#)
- [Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points](#)
- [Verwendung von Access Points mit mehreren Regionen mit unterstützten Vorgängen API](#)
- [Überwachung und Protokollierung von Anforderungen, die über einen Multi-Regions-Zugriffspunkt an zugrunde liegende Ressourcen erfolgen](#)

Berechtigungen

Amazon S3 Multiregion Access Points können den Datenzugriff für Amazon S3 S3-Buckets in mehreren Bereichen vereinfachen. AWS-Regionen Multi-Region Access Points sind benannte globale Endpunkte, mit denen Sie Datenzugriffsoperationen für Objekte in Amazon S3 ausführen können, z. B. `GetObject` und `PutObject`. Jeder Multi-Region Access Point kann für jede Anforderung, die über den globalen Endpunkt eingehen, über unterschiedliche Berechtigungen und Netzwerkkontrollen verfügen.

Jeder Multi-Region Access Point kann außerdem eine benutzerdefinierte Zugriffsrichtlinie erzwingen, die in Verbindung mit der Bucket-Richtlinie funktioniert, welche dem zugrunde liegenden Bucket angefügt ist. Damit eine Anforderung erfolgreich ist, müssen alle folgenden Komponenten die Operation zulassen:

- Die Richtlinie der Multi-Region Access Points
- Die zugrunde liegende Richtlinie AWS Identity and Access Management (IAM)
- Die zugrunde liegende Bucket-Richtlinie (an die die Anforderung weitergeleitet wird)

Sie können jede Access Point-Richtlinie für mehrere Regionen so konfigurieren, dass sie nur Anfragen von bestimmten IAM Benutzern oder Gruppen akzeptiert. Ein Beispiel für diese Vorgehensweise finden Sie in Beispiel 2 unter [the section called “Beispielrichtlinien für Multi-Region Access Points”](#). Um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk zu beschränken, können Sie die Multi-Region-Zugriffspunkt-Richtlinie so konfigurieren, dass nur Anfragen von einer virtuellen privaten Cloud akzeptiert werden (VPC).

Nehmen wir zum Beispiel an, dass Sie eine `GetObject` Anfrage über einen Multi-Region-Access Point stellen, indem Sie einen Benutzer verwenden, der `AppDataReader` in Ihrem AWS Konto angerufen wird. Um sicherzustellen, dass die Anforderung nicht abgelehnt wird, muss der Benutzer `AppDataReader` die `s3:GetObject`-Berechtigung vom Multi-Region Access Point und von jedem Bucket, der dem Multi-Region Access Point zugrunde liegt, erhalten. `AppDataReader` kann keine Daten aus einem Bucket abrufen, der diese Berechtigung nicht erteilt.

Important

Das Delegieren der Zugriffskontrolle für einen Bucket an eine Multi-Region-Access Point-Richtlinie ändert nichts am Verhalten des Buckets, wenn direkt über seinen Bucket-Namen oder Amazon-Ressourcennamen () ARN auf den Bucket zugegriffen wird. Alle Operationen, die direkt für den Bucket ausgeführt werden, funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Richtlinie für Multi-Region Access Points einschließen, gelten nur für Anforderungen, die über diesen Multi-Region Access Point eingehen.

Verwalten des öffentlichen Zugriffs auf einen Multi-Regions-Zugriffspunkt

Multi-Regions-Zugriffspunkte unterstützen unabhängige Block-Einstellungen des öffentlichen Zugriffs für jeden Multi-Regions-Zugriffspunkt. Wenn Sie einen Multi-Regions-Zugriffspunkt erstellen, können Sie die Block-Einstellungen des öffentlichen Zugriffs für diesen Multi-Regions-Zugriffspunkt festlegen.

Note

Alle Einstellungen für „Öffentlichen Zugriff blockieren“, die unter „Einstellungen „Öffentlichen Zugriff beschränken“ für dieses Konto (in Ihrem eigenen Konto) oder Einstellungen „Öffentlichen Zugriff beschränken“ für externe Buckets aktiviert sind, gelten auch dann, wenn die unabhängigen Einstellungen für „Öffentlichen Zugriff blockieren“ für Ihren Multi-Region Access Point deaktiviert sind.

Für jede Anforderung, die über einen Multi-Region Access Point erfolgt, wertet Amazon S3 die Einstellungen zum Blockieren des öffentlichen Zugriffs für Folgendes aus:

- Den Multi-Region Access Point
- Die zugrunde liegenden Buckets (einschließlich externer Buckets)
- Das Konto, das den Multi-Region Access Point besitzt

- Das Konto, das die zugrunde liegenden Buckets besitzt (einschließlich externer Konten)

Wenn eine dieser Einstellungen darauf hinweist, dass die Anforderung gesperrt werden soll, lehnt Amazon S3 die Anforderung ab. Weitere Hinweise zur Funktion Amazon S3 Block Public Access finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Important

Alle Einstellungen für die Blockierung des öffentlichen Zugriffs sind für neue Multi-Region Access Points standardmäßig aktiviert. Sie müssen alle Einstellungen, die Sie nicht auf einen Multi-Regions-Zugriffspunkt anwenden möchten, explizit deaktivieren.

Nach dem Erstellen eines Multi-Region Access Point können Sie die Block-Public-Access-Einstellungen für den Access Point nicht mehr ändern.

Anzeigen der Einstellungen für die Blockierung des öffentlichen Zugriffs für einen Multi-Region Access Point

So zeigen Sie die Einstellungen für die Blockierung des öffentlichen Zugriffs für einen Multi-Region Access Point an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Points aus, den Sie überprüfen möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Überprüfen Sie unter Block Public Access settings for this Multi-Region Access Point (Einstellungen für die Blockierung des öffentlichen Zugriffs für diesen Multi-Region Access Point) die Einstellungen für die Blockierung des öffentlichen Zugriffs für Ihren Multi-Region Access Point.

Note

Nachdem der Multi-Region Access Point erstellt wurde, können Sie die Einstellungen für die Blockierung des öffentlichen Zugriffs nicht mehr bearbeiten. Wenn Sie den öffentlichen Zugriff blockieren möchten, stellen Sie daher sicher, dass Ihre Anwendungen

ohne öffentlichen Zugriff ordnungsgemäß funktionieren, bevor Sie einen Multi-Region Access Point erstellen.

Verwenden einer Richtlinie für Multi-Region Access Points

Die folgende Beispielrichtlinie für Access Points für mehrere Regionen gewährt einem IAM Benutzer Zugriff auf das Auflisten und Herunterladen von Dateien von Ihrem Multi-Region-Access Point. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
        "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/*"
      ]
    }
  ]
}
```

Verwenden Sie den folgenden Befehl `put-multi-region-access-point-policy`, um die Richtlinie für Multi-Region Access Points über die AWS Command Line Interface (AWS CLI) mit dem angegebenen Multi-Region Access Point zu verknüpfen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen. Jeder Multi-Region Access Point kann nur über eine Richtlinie verfügen. Daher ersetzt eine an `put-multi-region-access-point-policy` gerichtete Anforderung jede vorhandene Richtlinie, die dem angegebenen Multi-Region Access Point zugeordnet ist.

AWS CLI

```
aws s3control put-multi-region-access-point-policy
--account-id 111122223333
--details { "Name": "amzn-s3-demo-bucket-MultiRegionAccessPoint",
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
  \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam:111122223333:root
  \", \"Action\": [\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
  [ \"arn:aws:s3:111122223333:accesspoint/MultiRegionAccessPoint_alias\",
  \"arn:aws:s3:111122223333:accesspoint/MultiRegionAccessPoint_alias/object/*
  \"] } } }" }
```

Verwenden Sie den folgenden Befehl, um die Ergebnisse der vorherigen Operation abzufragen:

AWS CLI

```
aws s3control describe-multi-region-access-point-operation
--account-id 111122223333
--request-token-arn requestArn
```

Verwenden Sie den folgenden Befehl, um Ihre Richtlinie für Multi-Region Access Points abzurufen:

AWS CLI

```
aws s3control get-multi-region-access-point-policy
--account-id 111122223333
--name=amzn-s3-demo-bucket-MultiRegionAccessPoint
```

Bearbeiten der Richtlinie für Multi-Region Access Points

Die Multi-Region-Zugriffspunkt-Richtlinie (beschrieben in JSON) ermöglicht den Speicherzugriff auf die Amazon S3 S3-Buckets, die mit diesem Multi-Region-Access Point verwendet werden. Sie können bestimmten Prinzipalen die Ausführung verschiedener Aktionen auf Ihrem Multi-Region Access Point erlauben oder verweigern. Wenn eine Anforderung über den Multi-Region Access Point an einen Bucket weitergeleitet wird, gelten sowohl die Zugriffsrichtlinien für den Multi-Region Access Point als auch den Bucket. Die restriktivere Zugriffsrichtlinie hat immer Vorrang.

Note

Wenn ein Bucket Objekte enthält, die anderen Konten gehören, gilt die Richtlinie für Multi-Region Access Points nicht für Objekte, die anderen AWS-Konten gehören.

Nachdem Sie eine Richtlinie für Multi-Region Access Points angewendet haben, kann die Richtlinie nicht mehr gelöscht werden. Sie können entweder die Richtlinie bearbeiten oder eine neue Richtlinie erstellen, die die bestehende überschreibt.

So bearbeiten Sie die Richtlinie für Multi-Region Access Points

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Point aus, für den Sie die Richtlinie bearbeiten möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Scrollen Sie nach unten zum Abschnitt Richtlinie für Multi-Region Access Points. Wählen Sie Bearbeiten, um die Richtlinie zu aktualisieren (inJSON).
6. Die Seite Edit Multi-Region Access Point policy (Richtlinie für Multi-Region Access Points bearbeiten) wird angezeigt. Sie können die Richtlinie entweder direkt in das Textfeld eingeben oder Add statement (Anweisung hinzufügen) auswählen, um Richtlinienelemente aus einer Dropdownliste auszuwählen.

Note

Die Konsole zeigt automatisch den Amazon Resource Name (ARN) für den multiregionalen Access Point an, den Sie in der Richtlinie verwenden können. Für Beispielrychtlinien für Multi-Region Access Points vgl. [the section called "Beispielrichtlinien für Multi-Region Access Points"](#).

Beispielrichtlinien für Multi-Region Access Points

Amazon S3 Multiregion Access Points unterstützen Ressourcenrichtlinien AWS Identity and Access Management (IAM). Mithilfe dieser Richtlinien können Sie die Verwendung des Multi-Region Access Point nach Ressource, Benutzer oder anderen Bedingungen steuern. Damit eine Anwendung oder ein Benutzer über einen Multi-Region Access Point auf Objekte zugreifen kann, müssen sowohl der Multi-Region Access Point Zugriffspunkt als auch der zugrunde liegende Bucket den gleichen Zugriff erlauben.

Gehen Sie folgendermaßen vor, um den gleichen Zugriff sowohl auf den Multi-Region Access Point als auch auf den zugrunde liegenden Bucket zu erlauben:

- (Empfohlen) Wenn Sie die Zugriffskontrollen bei der Verwendung eines Amazon S3 Multi-Region Access Points vereinfachen möchten, delegieren Sie die Zugriffskontrolle für den Amazon-S3-Bucket an den Multi-Region Access Point. Ein Beispiel für diese Vorgehensweise finden Sie in Beispiel 1 in diesem Abschnitt.
- Fügen Sie der Richtlinie des zugrunde liegenden Buckets dieselben Berechtigungen hinzu, die in der Richtlinie des Multi Region Access Points enthalten sind.

Important

Das Delegieren der Zugriffskontrolle für einen Bucket an eine Multi-Region-Access Point-Richtlinie ändert nichts am Verhalten des Buckets, wenn direkt über seinen Bucket-Namen oder Amazon-Ressourcennamen () ARN auf den Bucket zugegriffen wird. Alle Operationen, die direkt für den Bucket ausgeführt werden, funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Richtlinie für Multi-Region Access Points einschließen, gelten nur für Anforderungen, die über diesen Multi-Region Access Point eingehen.

Example 1 – Delegieren des Zugriffs auf bestimmte Multi-Region Access Points in Ihrer Bucket-Richtlinie (für dasselbe Konto oder kontoübergreifend)

Das folgende Beispiel für eine Bucket-Richtlinie gewährt vollständigen Bucket-Zugriff auf einen bestimmten Multi-Region Access Point. Dies bedeutet, dass der gesamte Zugriff auf diesen Bucket durch die Richtlinien gesteuert wird, die dem Multi-Region Access Point angefügt sind. Wir empfehlen, Ihre Buckets auf diese Weise für alle Anwendungsfälle zu konfigurieren, die keinen direkten Zugriff auf den Bucket erfordern. Sie können diese Bucket-Richtlinienstruktur für Multi-Region Access Points in demselben Konto oder einem anderen Konto verwenden.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Principal" : { "AWS": "*" },
      "Action" : "*",
      "Resource" : [ "Bucket ARN", "Bucket ARN/*" ],
      "Condition": {
        "StringEquals" : { "s3:DataAccessPointArn" : "MultiRegionAccessPoint_ARN" }
      }
    }
  ]
}
```

Note

Wenn Sie für mehrere Multi-Region Access Points Zugriff erteilen, stellen Sie sicher, dass Sie jeden einzelnen Multi-Region Access Point auflisten.

Example 2 – Gewähren des Zugriffs auf einen Multi-Region Access Point für ein Konto in Ihrer Richtlinie für Multi-Region Access Points

Die folgende Richtlinie für multiregionale Access Points ermöglicht es dem Konto *123456789012*, die Objekte aufzulisten und zu lesen, die sich in dem multiregionalen Access Point befinden, der durch den *MultiRegionAccessPoint_ARN*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "MultiRegionAccessPoint_ARN",

```

```

        "MultiRegionAccessPoint_ARN/object/*"
    ]
}
]
}

```

Example 3 – Richtlinie für Multi-Region Access Points, die eine Bucket-Auflistung ermöglicht

Die folgende Richtlinie für multiregionale Access Points ermöglicht es

Benutzerkonten `123456789012`, die Objekte aufzulisten, die in dem multiregionalen Access Point enthalten sind, definiert durch `MultiRegionAccessPoint_ARN`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
      },
      "Action": "s3:ListBucket",
      "Resource": "MultiRegionAccessPoint_ARN"
    }
  ]
}

```

Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts

Multi-Regions-Zugriffspunkte in Amazon S3 haben die folgenden Einschränkungen und Beschränkungen.

- Namen für Multi-Regions-Zugriffspunkte:
 - Muss innerhalb eines einzelnen AWS Kontos eindeutig sein
 - Muss mit einer Zahl oder einem Kleinbuchstaben beginnen
 - Muss zwischen 3 und 50 Zeichen lang sein.
 - Dürfen nicht mit einem Bindestrich (-) beginnen oder enden
 - Dürfen keine Unterstriche (_), Großbuchstaben oder Punkte (.) enthalten
 - Können nicht mehr bearbeitet werden, nachdem sie erstellt wurden

- Aliase für Multi-Region Access Points werden von Amazon S3 generiert und können nicht bearbeitet oder wiederverwendet werden.
- Sie können über einen Multi-Region Access Point nicht mit Gateway-Endpunkten auf Daten zugreifen. Sie können jedoch über einen Multi-Region Access Point mit Schnittstellenendpunkten auf Daten zugreifen. Zur Verwendung AWS PrivateLink müssen Sie VPC-Endpoints erstellen. Weitere Informationen finden Sie unter [Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink](#).
- Um Multi-Region Access Points mit Amazon zu verwenden CloudFront, müssen Sie den Multi-Region Access Point als Custom Origin Vertriebstyp konfigurieren. Weitere Informationen zu verschiedenen Herkunftstypen finden Sie unter [Verwenden verschiedener Herkunft bei CloudFront](#) Verteilungen. Weitere Informationen zur Verwendung von Multi-Region-Access Points mit Amazon CloudFront finden Sie im Storage-Blog unter [Aufbau einer Active-Active-Proximity-basierten Anwendung in mehreren Regionen](#).AWS
- Mindestanforderungen für Multi-Regions-Zugriffspunkte:
 - Transport Layer Security (TLS) v1.2
 - Signatur-Version 4 (SigV4A)

Multi-Regions-Zugriffspunkte unterstützen Signatur-Version 4A. Diese Version von SigV4 ermöglicht das Signieren von Anforderungen für mehrere AWS-Regionen. Dies ist nützlich bei API-Operationen, die zu Datenzugriff von einer oder mehreren Regionen führen können. Wenn Sie ein AWS SDK verwenden, geben Sie Ihre Anmeldeinformationen an, und für Anfragen an Access Points mit mehreren Regionen wird Signature Version 4A ohne zusätzliche Konfiguration verwendet. Stellen Sie sicher, dass Sie die [Kompatibilität Ihres AWS SDK](#) mit dem SigV4A-Algorithmus überprüfen. Weitere Informationen zu SigV4a finden Sie unter [Signieren von AWS API-Anfragen](#) im. Allgemeine AWS-Referenz

Note

Um SigV4a mit temporären Sicherheitsanmeldedaten zu verwenden — beispielsweise bei der Verwendung von AWS Identity and Access Management (IAM-) Rollen — können Sie die temporären Anmeldeinformationen von einem regionalen () Endpunkt anfordern. AWS Security Token Service AWS STS Wenn Sie temporäre Anmeldeinformationen vom globalen AWS STS Endpunkt (sts.amazonaws.com) anfordern, müssen Sie zunächst die Regionskompatibilität der Sitzungstoken für den globalen Endpunkt so

einstellen, dass sie für alle gültig ist. AWS-Regionen Weitere Informationen finden Sie unter [Managing AWS STS AWS-Region in an](#) im IAM-Benutzerhandbuch.

- Multi-Region Access Points unterstützen keine anonymen Anforderungen.
- Beschränkungen des Multi-Regions-Zugriffspunkts:
 - IPv6 wird nicht unterstützt.
 - Buckets von Amazon S3 on Outposts werden nicht unterstützt.
 - Multi-Region Access Points unterstützen Kopiervorgänge, bei denen Multi-Region Access Points nur als Ziel verwendet werden, wenn der Multi-Region Access Point ARN verwendet wird.
 - Die Funktion S3 Batch Operations wird nicht unterstützt.
- Bestimmte AWS SDKs werden nicht unterstützt. Informationen darüber, welche AWS SDKs für Access Points mit mehreren Regionen unterstützt werden, finden Sie unter [Kompatibilität mit SDKs](#). AWS
- Service Quotas für Multi-Region Access Points lauten wie folgt:
 - Pro Konto gibt es maximal 100 Multi-Regions-Zugriffspunkte.
 - Es gibt ein Limit von 17 Regionen für einen einzelnen Multi-Region Access Point.
- Nach der Erstellung eines Multi-Region Access Point können Sie keine Buckets hinzufügen, ändern oder aus der Konfiguration des Multi-Region Access Point entfernen. Um die Buckets zu ändern, müssen Sie den gesamten Multi-Regions-Zugriffspunkt löschen und einen neuen erstellen. Wenn ein kontoübergreifender Bucket in Ihrem Multi-Region Access Point gelöscht wird, besteht die einzige Möglichkeit, diesen Bucket erneut zu verbinden, darin, den Bucket neu zu erstellen und dabei denselben Namen und dieselbe Region in diesem Konto zu verwenden.
- Zugrunde liegende Buckets (in demselben Konto), die in einem Multi-Region Access Point verwendet werden, können erst nach dem Löschen eines Multi-Region Access Point gelöscht werden.
- Alle Anforderungen auf Steuerebene zum Erstellen oder Verwalten von Multi-Region Access Points müssen an die Region US West (Oregon) weitergeleitet werden. Für Anforderungen von Multi-Region Access Point auf Steuerebene müssen keine Regionen angegeben werden.
- Für die Failover-Stuerebene von Multi-Region Access Points muss die Anforderung an eine dieser fünf unterstützten Regionen weitergeleitet werden:
 - US East (N. Virginia)
 - US West (Oregon)
 - Asia Pacific (Sydney)

- Asia Pacific (Tokyo)
- Europe (Ireland)
- Ihr Access Point für mehrere Regionen unterstützt nur Buckets in den folgenden Bereichen: AWS-Regionen
 - US East (N. Virginia)
 - US East (Ohio)
 - US West (N. California)
 - US West (Oregon)
 - Asia Pacific (Mumbai)
 - Asia Pacific (Osaka)
 - Asia Pacific (Seoul)
 - Asia Pacific (Singapore)
 - Asia Pacific (Sydney)
 - Asia Pacific (Tokyo)
 - Canada (Central)
 - Europe (Frankfurt)
 - Europe (Ireland)
 - Europe (London)
 - Europe (Paris)
 - Europe (Stockholm)
 - South America (São Paulo)

Weiterleitung der Multi-Regions-Zugriffspunktanforderung

Wenn Sie eine Anforderung über einen Multi-Region Access Point stellen, bestimmt Amazon S3, welche der Buckets, die dem Multi-Region Access Point zugeordnet sind, sich in Ihrer nächster Nähe befinden. Amazon S3 leitet die Anforderung dann an diesen Bucket weiter, unabhängig von der AWS-Region, in der sie sich befindet.

Nachdem der Multi-Region Access Point die Anforderung an den nächstgelegenen Bucket weitergeleitet hat, verarbeitet Amazon S3 die Anforderung so, als ob Sie sie direkt an diesen Bucket gestellt hätten. Multi-Region Access Points kennen den Dateninhalt eines Amazon-S3-Buckets nicht. Daher kann es sein, dass der Bucket, der die Anforderung erhält, die angeforderten Daten nicht

enthält. Um konsistente Datensätze in den Amazon-S3-Buckets zu schaffen, die einem Multi-Region Access Point zugeordnet sind, können Sie S3 Cross-Region Replication (CRR, regionsübergreifende Replikation) konfigurieren. Dann kann jeder Bucket die -Anforderung erfolgreich erfüllen.

Amazon S3 leitet Multi-Regions-Zugriffspunktanforderungen gemäß den folgenden Regeln weiter:

- Amazon S3 optimiert Anforderungen, um der Nähe entsprechend erfüllt zu werden. Es untersucht die Buckets, die vom Multi-Region Access Point unterstützt werden, und leitet die Anforderung an den Bucket weiter, der sich in nächster Nähe befindet.
- Wenn die Anforderung eine vorhandene Ressource angibt (z. B. `GetObject`), berücksichtigt Amazon S3 nicht den Namen des Objekts bei der Erfüllung der Anforderung. Selbst wenn ein Objekt in einem Bucket in dem Multi-Region Access Point vorhanden ist, kann Ihre Anforderung somit an einen Bucket weitergeleitet werden, der das Objekt nicht enthält. Dies führt dazu, dass dem Kunden eine 404-Fehlermeldung zurückgegeben wird.

Um 404-Fehler zu vermeiden, empfehlen wir Ihnen, S3 Cross-Region Replication (CRR) für Ihre Buckets zu konfigurieren. Die Replikation hilft bei der Lösung des potenziellen Problems, das entsteht, wenn sich das gewünschte Objekt in einem Bucket in dem Multi-Region Access Point befindet, nicht jedoch in dem bestimmten Bucket, an den Ihre Anforderung weitergeleitet wurde. Weitere Informationen zum Erstellen einer grundlegenden Replikationskonfiguration finden Sie unter [Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points](#).

Um sicherzustellen, dass Ihre Anforderungen mithilfe der gewünschten Objekte erfüllt werden, wird außerdem empfohlen, die Bucket-Versionsverwaltung zu aktivieren und Versions-IDs in Ihre Anforderungen aufzunehmen. Auf diese Weise stellen Sie sicher, dass Sie über die richtige Version des gesuchten Objekts verfügen. Buckets mit aktivierter Versionsverwaltung können auch bei der Wiederherstellung von Objekten nach einem versehentlichen Überschreiben hilfreich sein. Weitere Informationen finden Sie unter [Verwenden des S3-Versioning in S3-Buckets](#).

- Wenn sich die Anforderung auf das Erstellen einer Ressource bezieht (z. B. `PutObject` oder `CreateMultipartUpload`), erfüllt Amazon S3 die Anforderung unter Verwendung des nächstgelegenen Buckets. Betrachten wir beispielsweise ein Videounternehmen, das Video-Uploads von überall auf der Welt unterstützen möchte. Wenn ein Benutzer eine PUT-Anforderung an den Multi-Region Access Point stellt, wird das Objekt in den nächstgelegenen Bucket gestellt. Um das hochgeladene Video dann anderen auf der ganzen Welt mit der geringsten Latenz zum Download zur Verfügung zu stellen, können Sie CRR mit bidirektionaler Replikation verwenden. Wenn Sie CRR mit bidirektionaler Replikation verwenden, bleiben die Inhalte aller Buckets, die dem Multi-Region Access Point zugeordnet sind, synchronisiert. Weitere Informationen zur

Verwendung der Replikation mit Multi-Region Access Points finden Sie unter [Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points](#).

Failover-Kontrollen für Amazon S3 Multi-Region Access Points

Mit den Failover-Kontrollen für Amazon S3 Multi-Region Access Points können Sie die Geschäftskontinuität bei regionalen Verkehrsunterbrechungen aufrechterhalten und Ihren Anwendungen gleichzeitig eine regionsübergreifende Architektur zur Erfüllung von Compliance- und Redundanzanforderungen bieten. Wenn Ihr regionaler Datenverkehr unterbrochen wird, können Sie mithilfe der Failover-Kontrollen für Multi-Region Access Points auswählen, welche AWS-Regionen hinter einem Amazon S3 Multi-Region Access Point Datenzugriffs- und Speicheranforderungen verarbeiten.

Zur Failover-Unterstützung können Sie Ihren Multi-Region Access Point in einer Aktiv-Passiv-Konfiguration einrichten, wobei der Datenverkehr unter normalen Bedingungen in die aktive Region fließt und sich eine passive Region für den Failover im Standby-Modus befindet.

Wenn Sie beispielsweise ein Failover auf eine AWS-Region Ihrer Wahl durchführen möchten, verlagern Sie den Datenverkehr von Ihrer primären (aktiven) Region in Ihre sekundäre (passive) Region. In einer solchen Aktiv-Passiv-Konfiguration ist ein Bucket aktiv und akzeptiert Datenverkehr, während der andere Bucket passiv ist und keinen Traffic akzeptiert. Der passive Bucket wird für die Notfallwiederherstellung verwendet. Wenn Sie den Failover einleiten, wird der gesamte Datenverkehr (wie GET- und PUT-Anforderungen) an den Bucket im aktiven Zustand (in einer Region) und weg vom Bucket im passiven Zustand (in einer anderen Region) geleitet.

Wenn Sie die regionsübergreifende Replikation in S3 (CRR) mit bidirektionalen Replikationsregeln aktiviert haben, können Sie Ihre Buckets während eines Failovers synchronisieren. Wenn Sie CRR in einer Aktiv-Aktiv-Konfiguration aktiviert haben, können Amazon S3 Multi-Region Access Points außerdem Daten von dem Bucket-Standort in nächster Nähe abrufen, was die Anwendungsleistung verbessert.

AWS-Region-Support

Mit den Failover-Kontrollen für Amazon S3 Multi-Region Access Points können sich Ihre S3-Buckets in jeder der [17 Regionen](#) befinden, in denen Multi-Region Access Points unterstützt werden. Sie können ein Failover für zwei beliebige Regionen gleichzeitig initiieren.

Note

Obwohl ein Failover nur zwischen zwei Regionen gleichzeitig initiiert wird, können Sie den Routing-Status für mehrere Regionen gleichzeitig in Ihrem Multi-Region Access Point separat aktualisieren.

In den folgenden Themen wird die Verwendung und Verwaltung von Failover-Kontrollen für Amazon S3 Multi-Region Access Points veranschaulicht.

Themen

- [Weiterleitungsstatus von Amazon S3 Multi-Region Access Points](#)
- [Verwenden von Failover-Kontrollen für Amazon S3 Multi-Region Access Points](#)
- [Failover-Kontrollfehler für Amazon S3 Multi-Region Access Points](#)

Weiterleitungsstatus von Amazon S3 Multi-Region Access Points

Ihre Failover-Konfiguration für Amazon S3 Multi-Region Access Points bestimmt den Weiterleitungsstatus der AWS-Regionen, die mit dem Multi-Region Access Point verwendet werden. Sie können Ihren Amazon S3 Multi-Region Access Point so konfigurieren, dass er sich in einem Aktiv-Aktiv- oder Aktiv-Passiv-Zustand befindet.

- **Aktiv-Aktiv** – In einer Aktiv-Aktiv-Konfiguration werden alle Anfragen automatisch an die nächstgelegenen AWS-Region in Ihrem Multi-Region Access Point gesendet. Nachdem der Multi-Region Access Point so konfiguriert wurde, dass er sich in einem Aktiv-Aktiv-Zustand befindet, können alle Regionen Datenverkehr empfangen. Wenn in einer Aktiv-Aktiv-Konfiguration eine Verkehrsunterbrechung auftritt, wird der Netzwerkverkehr automatisch in eine der aktiven Regionen umgeleitet.
- **Aktiv-Passiv** – In einer Aktiv-Passiv-Konfiguration empfangen die aktiven Regionen in Ihrem Multi-Region Access Point Traffic und die passiven Regionen nicht. Wenn Sie beabsichtigen, S3-Failover-Kontrollen zu verwenden, um in einer Notfallsituation ein Failover einzuleiten, richten Sie Ihre Multi-Region Access Points in einer Aktiv-Passiv-Konfiguration ein, während Sie die Notfallwiederherstellungsplanung testen und durchführen.

Verwenden von Failover-Kontrollen für Amazon S3 Multi-Region Access Points

In diesem Abschnitt wird erläutert, wie Sie Ihre Amazon S3 Multi-Region Access Points mithilfe der AWS Management Console verwalten und verwenden.

Auf der Detailseite Ihres Multi-Region Access Points finden Sie im Abschnitt Failover configuration (Failover-Konfiguration) in der AWS Management Console zwei Failover-Kontrollen: Edit routing status (Weiterleitungsstatus bearbeiten) und Failover. Verwenden Sie diese Steuerelemente wie folgt:

- Edit routing status (Weiterleitungsstatus bearbeiten) – Sie können den Weiterleitungsstatus von bis zu 17 AWS-Regionen in einer einzigen Anforderung für Ihren Multi-Region Access Point manuell bearbeiten, indem Sie Edit routing status (Weiterleitungsstatus bearbeiten) auswählen. Sie können die Option Edit routing status (Weiterleitungsstatus bearbeiten) für folgende Zwecke verwenden:
 - Zum Festlegen oder Bearbeiten der Weiterleitungsstatus einer oder mehrerer Regionen in Ihrem Multi-Region Access Point
 - Zum Erstellen einer Failover-Konfiguration für Ihren Multi-Region Access Point, indem Sie zwei Regionen so konfigurieren, dass sie sich in einem Aktiv-Passiv-Zustand befinden
 - So führen Sie ein manuelles Failover für Ihre Regionen durch
 - So leiten Sie Traffic manuell zwischen Regionen um
- Failover – Wenn Sie ein Failover initiieren, indem Sie Failover auswählen, aktualisieren Sie nur den Weiterleitungsstatus von zwei Regionen, die bereits so konfiguriert sind, dass sie sich in einem Aktiv-Passiv-Zustand befinden. Während eines Failovers, das Sie initiiert haben, indem Sie Failover auswählen, wird der Weiterleitungsstatus zwischen den beiden Regionen automatisch gewechselt.

Bearbeiten des Weiterleitungsstatus der Regionen in Ihrem Multi-Region Access Point

Sie können den Weiterleitungsstatus von bis zu 17 AWS-Regionen in einer einzigen Anforderung für Ihren Multi-Region Access Point manuell aktualisieren, indem Sie auf der Detailseite Ihres Multi-Region Access Points im Abschnitt Failover configuration (Failover-Konfiguration) die Option Edit routing status (Weiterleitungsstatus bearbeiten) auswählen. Wenn Sie jedoch ein Failover initiieren, indem Sie Failover auswählen, aktualisieren Sie nur den Weiterleitungsstatus von zwei Regionen, die bereits so konfiguriert sind, dass sie sich in einem Aktiv-Passiv-Zustand befinden. Während eines Failovers, das Sie initiiert haben, indem Sie Failover auswählen, wird der Weiterleitungsstatus zwischen den beiden Regionen automatisch gewechselt.

Sie können die Option Edit routing status (Weiterleitungsstatus bearbeiten) (wie im folgenden Verfahren beschrieben) für die folgenden Zwecke verwenden:

- Zum Festlegen oder Bearbeiten der Weiterleitungsstatus einer oder mehrerer Regionen in Ihrem Multi-Region Access Point
- Zum Erstellen einer Failover-Konfiguration für Ihren Multi-Region Access Point, indem Sie zwei Regionen so konfigurieren, dass sie sich in einem Aktiv-Passiv-Zustand befinden
- So führen Sie ein manuelles Failover für Ihre Regionen durch
- So leiten Sie Traffic manuell zwischen Regionen um

Verwenden der S3-Konsole

So aktualisieren Sie den Weiterleitungsstatus der Regionen in Ihrem Multi-Region Access Point

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
3. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
4. Wählen Sie den Multi-Region Access Point aus, den Sie aktualisieren möchten.
5. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
6. Wählen Sie eine oder mehrere Regionen aus, deren Weiterleitungsstatus Sie bearbeiten möchten.

Note

Damit ein Failover initiiert wird, muss mindestens eine AWS-Region in Ihrem Multi-Region Access Point als Active (Aktiv) und eine Region als Passive (Passiv) festgelegt sein.

7. Wählen Sie Edit routing status (Weiterleitungsstatus bearbeiten) aus.
8. Wählen Sie in dem daraufhin angezeigten Dialogfeld Active (Aktiv) oder Passive (Passiv) als Routing status (Weiterleitungsstatus) für jede Region aus.

Ein aktiver Status ermöglicht die Weiterleitung von Verkehr an die Region. Ein passiver Status verhindert, dass jeglicher Verkehr an die Region weitergeleitet wird.

Wenn Sie eine Failover-Konfiguration für Ihren Multi-Region Access Point erstellen oder ein Failover initiieren, muss mindestens eine AWS-Region in Ihrem Multi-Region Access Point als Active (Aktiv) und eine Region als Passive (Passiv) festgelegt sein.

9. Wählen Sie Save routing status (Weiterleitungssatus speichern) aus. Es dauert etwa 2 Minuten, bis der Verkehr umgeleitet wird.

Nachdem Sie den Weiterleitungssatus der AWS-Regionen für Ihren Multi-Region Access Point übermittelt haben, können Sie die Änderungen des Weiterleitungssatus überprüfen. Zur Überprüfung dieser Änderungen rufen Sie Amazon CloudWatch unter <https://console.aws.amazon.com/cloudwatch> auf. Hier können Sie die Verlagerung Ihres Datenanforderungsverkehrs von Amazon S3 (z. B. GET- und PUT-Anforderungen) zwischen aktiven und passiven Regionen überwachen. Bestehende Verbindungen werden während des Failovers nicht beendet. Bestehende Verbindungen werden fortgesetzt, bis der Erfolgs- bzw. Misserfolgsstatus angezeigt wird.

Verwendung von AWS CLI

Note

Sie können AWS CLI-Weiterleitungsbefehle für den Multi-Region Access Point in einer dieser fünf Regionen ausführen:

- ap-southeast-2
- ap-northeast-1
- us-east-1
- us-west-2
- eu-west-1

Mit dem folgenden Beispielbefehl wird die aktuelle Weiterleitungskonfiguration Ihres Multi-Region Access Points aktualisiert. Wenn Sie den aktiven oder passiven Status eines Buckets aktualisieren möchten, legen Sie den Wert TrafficDialPercentage auf 100 für aktiv und auf 0 für passiv fest. In diesem Beispiel ist *DOC-EXAMPLE-BUCKET-1* auf aktiv und *DOC-EXAMPLE-BUCKET-2* auf passiv eingestellt. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control submit-multi-region-access-point-routes
```

```
--region ap-southeast-2
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
--route-updates Bucket=DOC-EXAMPLE-BUCKET-1,TrafficDialPercentage=100
                Bucket=DOC-EXAMPLE-BUCKET-2,TrafficDialPercentage=0
```

Mit dem folgenden Beispielbefehl wird die aktualisierte Weiterleitungskonfiguration Ihres Multi-Region Access Points abgerufen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
```

Initiieren eines Failovers

Wenn Sie einen Failover einleiten, indem Sie auf der Detailseite Ihres Multi-Region Access Points im Abschnitt Failover configuration (Failover-Konfiguration) die Option Failover auswählen, wird der Amazon-S3-Anforderungsverkehr automatisch auf eine alternative AWS-Region umgeleitet. Der Failover-Prozess ist innerhalb von 2 Minuten abgeschlossen.

Sie können ein Failover für zwei beliebige AWS-Regionen gleichzeitig initiieren (zwei der [17 Regionen](#), in denen Multi-Region Access Points unterstützt werden). Failover-Ereignisse werden dann in AWS CloudTrail protokolliert. Nach Abschluss des Failovers können Sie den Amazon-S3-Traffic und alle Aktualisierungen der Verkehrsweiterleitungen auf die neue aktive Region in Amazon CloudWatch überwachen.

Important


Damit alle Metadaten und Objekte während der Datenreplikation bucketübergreifend synchronisiert werden, empfehlen wir, bidirektionale Replikationsregeln zu erstellen und die Synchronisierung von Replikatänderungen zu aktivieren, bevor Sie Ihre Failover-Kontrollen konfigurieren.

Bidirektionale Replikationsregeln tragen dazu bei, sicherzustellen, dass wenn Daten in den Amazon-S3-Bucket geschrieben werden, auf den der Datenverkehr bei einem Failover zurückgreift, diese Daten dann zurück in den Quell-Bucket repliziert werden. Durch die Synchronisierung von Replikatänderungen wird sichergestellt, dass die Objektmetadaten während der bidirektionalen Replikation auch zwischen Buckets synchronisiert werden.

Weitere Informationen zum Konfigurieren der Replikation zur Unterstützung von Failover finden Sie unter [the section called "Bucket-Replikation"](#).

So leiten Sie ein Failover zwischen replizierten Buckets ein

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
3. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
4. Wählen Sie den Multi-Region Access Point aus, den Sie zum Initiieren des Failovers verwenden möchten.
5. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
6. Scrollen Sie nach unten zum Abschnitt Failover configuration (Failover-Konfiguration) und wählen Sie zwei AWS-Regionen aus.

 Note

Damit ein Failover initiiert wird, muss mindestens eine AWS-Region in Ihrem Multi-Region Access Point als Active (Aktiv) und eine Region als Passive (Passiv) festgelegt sein. Ein aktiver Status ermöglicht die Weiterleitung von Verkehr an eine Region. Ein passiver Status verhindert, dass jeglicher Verkehr an die Region weitergeleitet wird.

7. Klicken Sie auf Failover.
8. Wählen Sie im Dialogfeld erneut Failover aus, um den Failover-Vorgang einzuleiten. Während dieses Vorgangs werden die Weiterleitungsstatus der beiden Regionen automatisch gewechselt. Der gesamte neue Traffic wird an die Region geleitet, die aktiv wird, und der Verkehr wird nicht mehr an die Region weitergeleitet, die passiv wird. Es dauert etwa 2 Minuten, bis der Verkehr umgeleitet wird.

Nachdem Sie den Failover-Prozess eingeleitet haben, können Sie Ihre Verkehrsänderungen überprüfen. Zur Überprüfung dieser Änderungen rufen Sie Amazon CloudWatch unter <https://console.aws.amazon.com/cloudwatch> auf. Hier können Sie die Verlagerung Ihres Datenanforderungsverkehrs von Amazon S3 (z. B. GET- und PUT-Anforderungen) zwischen aktiven und passiven Regionen überwachen. Bestehende Verbindungen werden während des Failovers nicht beendet. Bestehende Verbindungen werden fortgesetzt, bis der Erfolgs- bzw. Misserfolgsstatus angezeigt wird.

Anzeigen Ihrer Weiterleitungskontrollen für Amazon S3 Multi-Region Access Points

Verwenden der S3-Konsole

So zeigen Sie Ihre Weiterleitungskontrollen für Ihren Amazon S3 Multi-Region Access Point an

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
3. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
4. Wählen Sie den Multi-Region Access Point aus, den Sie überprüfen möchten.
5. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus. Auf dieser Seite werden die Konfigurationsdetails für die Weiterleitung und eine Zusammenfassung für Ihren Multi-Region Access Point, die zugehörigen Replikationsregeln und Replikationsmetriken angezeigt. Sie können den Weiterleitungsstatus Ihrer Regionen im Abschnitt Failover configuration (Failover-Konfiguration) sehen.

Verwendung von AWS CLI

Mit dem folgenden AWS CLI-Beispielbefehl wird die aktuelle Weiterleitungskonfiguration Ihres Multi-Region Access Points für die angegebene Region abgerufen. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrp MultiRegionAccessPoint_ARN
```

Note

Dieser Befehl kann nur für diese fünf Regionen ausgeführt werden:

- *ap-southeast-2*
- *ap-northeast-1*
- *us-east-1*
- *us-west-2*
- *eu-west-1*

Failover-Kontrollfehler für Amazon S3 Multi-Region Access Points

Wenn Sie die Failover-Konfiguration für Ihren Multiregion Access Point aktualisieren, tritt möglicherweise einer der folgenden Fehler auf:

- **HTTP 400 Ungültige Anforderung:** Dieser Fehler kann auftreten, wenn Sie bei der Aktualisierung Ihrer Failover-Konfiguration einen ungültigen ARN für den Multi-Region Access Point eingeben. Sie können den ARN Ihres Multi-Region Access Points anhand der Richtlinie des Multi-Region Access Points überprüfen. Informationen zur Überprüfung oder Aktualisierung Ihrer der Richtlinie Ihres Multi-Region Access Points finden Sie unter [Bearbeiten der Richtlinie für den Multi-Region Access Point](#). Dieser Fehler kann auch auftreten, wenn Sie beim Aktualisieren der Failover-Kontrollen für Ihren Amazon S3 Multi-Region Access Point eine leere oder zufällige Zeichenfolge verwenden. Für ARNs von Multi-Region Access Points muss folgendes Format verwendet werden:

```
arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias
```

- **HTTP 503 Slow Down:** Dieser Fehler tritt auf, wenn Sie innerhalb kurzer Zeit zu viele Anforderungen senden. Abgelehnte Anforderungen führen zu einem Fehler.
- **HTTP 409 Conflict:** Dieser Fehler tritt auf, wenn zwei oder mehr gleichzeitige Aktualisierungsanforderungen für die Weiterleitungskonfiguration an einen einzelnen Multi-Region Access Point gerichtet sind. Die erste Anforderung ist erfolgreich, aber alle anderen Anforderungen schlagen mit einem Fehler fehl.
- **HTTP 405 Method Not Allowed:** Dieser Fehler tritt auf, wenn Sie beim Initiieren des Failovers einen Multi-Region Access Point mit nur einer AWS-Region ausgewählt haben. Sie müssen zwei Regionen auswählen, bevor Sie den Failover einleiten können. Andernfalls wird ein Fehler zurückgegeben.

Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points

Wenn Sie eine Anforderung an einen Endpunkt eines Multi-Region Access Point stellen, leitet Amazon S3 die Anforderung automatisch an den Ihnen nächstgelegenen Bucket weiter. Amazon S3 berücksichtigt die Inhalte der Anforderung bei dieser Entscheidung nicht. Wenn Sie eine Anforderung stellen, um ein Objekt mit GET abzurufen, wird Ihre Anforderung möglicherweise an einen Bucket weitergeleitet, der keine Kopie dieses Objekts besitzt. In diesem Fall erhalten Sie den HTTP-Statuscode 404 (Nicht gefunden). Weitere Informationen zur Weiterleitung von Multi-Region-Access-Point-Anforderungen finden Sie unter [the section called "Weiterleitung von Anforderungen"](#).

Wenn Sie möchten, dass der Multi-Region Access Point das Objekt unabhängig davon wiederherstellen kann, welcher Bucket die Anforderung empfängt, müssen Sie die regionsübergreifende Replikation von Amazon S3 (CRR) konfigurieren.

Betrachten Sie einen Multi-Region Access Point mit drei Buckets:

- Ein Bucket mit dem Namen `my-bucket-usw2` in der Region `us-west-2`, der das Objekt `my-image.jpg` enthält
- Ein Bucket mit dem Namen `my-bucket-aps1` in der Region `ap-south-1`, der das Objekt `my-image.jpg` enthält
- Ein Bucket mit dem Namen `my-bucket-euc1` in der Region `eu-central-1`, der das Objekt `my-image.jpg` nicht enthält

Wenn Sie in dieser Situation eine `GetObject`-Anforderung für das Objekt `my-image.jpg` stellen, hängt der Erfolg dieser Anforderung davon ab, welcher Bucket Ihre Anforderung empfängt. Da Amazon S3 den Inhalt der Anforderung nicht berücksichtigt, kann es Ihre `GetObject`-Anforderung an den `my-bucket-euc1`-Bucket weiterleiten, wenn dieser Bucket aus nächster Nähe antwortet. Obwohl sich Ihr Objekt in einem Bucket im Multi-Region Access Point befindet, erhalten Sie einen HTTP-Fehler 404 Not Found, da der einzelne Bucket, der Ihre Anforderung empfangen hat, das Objekt nicht enthielt.

Durch das Aktivieren der regionsübergreifenden Replikationsfunktion (CRR) kann dieses Ergebnis vermieden werden. Mit den entsprechenden Replikationsregeln wird das `my-image.jpg`-Objekt in den `my-bucket-euc1`-Bucket kopiert. Wenn Amazon S3 Ihre Anforderung an diesen Bucket weiterleitet, können Sie das Objekt daher jetzt abrufen.

Die Replikation funktioniert normal mit Buckets, die einem Multi-Regions-Zugriffspunkt zugewiesen sind. Amazon S3 führt keine spezielle Replikationsbehandlung mit Buckets aus, die sich in Multi-Region Access Points befinden. Weitere Informationen zum Konfigurieren einer Replikation in Ihren Buckets finden Sie unter [Live-Replikation einrichten](#).

Empfehlungen zur Verwendung der Replikationsfunktion mit Multi-Region Access Points

Für beste Replikationsleistung bei der Arbeit mit Multi-Region Access Points empfehlen wir Folgendes:

- Konfigurieren Sie die Begrenzung der S3-Replikationszeit (S3 RTC). Um Ihre Daten innerhalb eines vorhersehbaren Zeitraums über verschiedene Regionen hinweg zu replizieren, können Sie S3 RTC verwenden. S3 RTC repliziert 99,99 Prozent der neuen in Amazon S3 gespeicherten

Objekte innerhalb von 15 Minuten (gestützt auf ein Service Level Agreement). Weitere Informationen finden Sie unter [the section called “Verwenden der S3-Replikationszeitkontrolle”](#). Für S3 RTC fallen zusätzliche Gebühren an. Informationen finden Sie unter [Amazon S3 – Preise](#).

- Verwenden Sie die bidirektionale Replikation, um die Synchronisierung von Buckets zu unterstützen, wenn Buckets über den Multi-Region Access Point aktualisiert werden. Weitere Informationen finden Sie unter [the section called “Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point”](#).
- Erstellen Sie kontoübergreifende Multi-Region Access Points für die Replikation von Daten in Buckets in separaten AWS-Konten. Dieses Vorgehen ermöglicht eine Trennung auf Kontoebene, sodass auf Daten von verschiedenen Konten in anderen Regionen als dem Quell-Bucket zugegriffen werden kann und diese repliziert werden können. Die Einrichtung von kontoübergreifenden Multi-Region Access Points ist ohne zusätzliche Kosten möglich. Wenn Sie Bucket-Eigentümer, jedoch nicht Eigentümer des Multi-Region Access Point sind, zahlen Sie nur für die Datenübertragung und die Anforderungskosten. Eigentümer von Multi-Region Access Points tragen die Kosten für die Datenweiterleitung und die Internetbeschleunigung. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).
- Aktivieren Sie die Synchronisierung von Replikatänderungen für jede Replikationsregel, um auch Metadatenänderungen an Ihren Objekten zu synchronisieren. Weitere Informationen finden Sie unter [Aktivieren der Synchronisierung von Replikatänderungen](#).
- Aktivieren Sie Amazon-CloudWatch-Metriken, um [Replikationsereignisse zu überwachen](#). Es fallen Gebühren für CloudWatch-Metriken an. Weitere Informationen hierzu finden Sie unter [Amazon CloudWatch – Preise](#).

Themen

- [Erstellen von unidirektionalen Replikationsregeln für Ihren Multi-Region Access Point](#)
- [Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point](#)
- [Anzeigen der Replikationsregeln für Ihren Multi-Region Access Point](#)

Erstellen von unidirektionalen Replikationsregeln für Ihren Multi-Region Access Point

Replikationsregeln ermöglichen ein automatisches und asynchrones Kopieren von Objekten über Buckets hinweg. Eine unidirektionale Replikationsregel hilft sicherzustellen, dass Daten vollständig aus einem Quell-Bucket in einer AWS-Region in einen Ziel-Bucket in einer anderen Region repliziert werden. Wenn unidirektionale Replikation eingerichtet ist, wird eine Replikationsregel vom Quell-Bucket (DOC-EXAMPLE-BUCKET-1) zum Ziel-Bucket (DOC-EXAMPLE-BUCKET-2) erstellt. Wie

alle Replikationsregeln können Sie die unidirektionale Replikationsregel entweder auf den gesamten Amazon-S3-Bucket oder auf eine anhand eines Präfixes oder von Objekt-Tags gefilterte Teilmenge von Objekten anwenden.

Important

Wir empfehlen eine unidirektionale Replikation, wenn Ihre Benutzer die Objekte in Ihren Ziel-Buckets nur verwenden werden. Wenn Ihre Benutzer die Objekte in Ihren Ziel-Buckets hochladen oder ändern, verwenden Sie die bidirektionale Replikation, damit alle Ihre Buckets synchronisiert bleiben. Wir empfehlen eine bidirektionale Replikation auch, wenn Sie Ihren Multi-Region Access Point für Failover verwenden möchten. Informationen zum Einrichten der bidirektionalen Replikation finden Sie unter [the section called “Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point”](#).

So erstellen Sie eine unidirektionale Replikationsregel für Ihren Multi-Region Access Point

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen Ihres Multi-Region Access Point aus.
4. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
5. Scrollen Sie nach unten zum Abschnitt Replication rules (Replikationsregeln) und wählen Sie dann Create replication rules (Replikationsregeln erstellen) aus. Vergewissern Sie sich, dass Sie über ausreichende Berechtigungen verfügen, um die Replikationsregel zu erstellen. Andernfalls wird die Versionsverwaltung deaktiviert.

Note

Sie können nur für Buckets in Ihrem eigenen Konto Replikationsregeln erstellen. Replikationsregeln für externe Buckets müssen die Bucket-Eigentümer selbst erstellen.

6. Wählen Sie auf der Seite Replikationsregeln erstellen die Vorlage zum Replizieren von Objekten aus einem oder mehreren Quell-Buckets in einen oder mehrere Ziel-Buckets aus.

⚠ Important

Wenn Sie Replikationsregeln mithilfe dieser Vorlage erstellen, ersetzen diese alle vorhandenen Replikationsregeln, die dem Bucket bereits zugewiesen sind.

Wenn Sie Replikationsregeln hinzufügen oder bestehende Regeln ändern möchten, anstatt sie zu ersetzen, wechseln Sie in der Konsole zur Registerkarte Management (Verwaltung) der einzelnen Buckets und bearbeiten Sie dann die Regeln im Abschnitt Replication rules (Replikationsregeln). Sie können bestehende Replikationsregeln auch mithilfe der AWS CLI, SDKs oder der REST-API hinzufügen oder ändern. Weitere Informationen finden Sie unter [Replikationskonfiguration](#).

7. Wählen Sie im Abschnitt Quelle und Ziel unter Quell-Buckets einen oder mehrere Buckets aus, aus dem/denen Sie Objekte replizieren möchten. Für alle für die Replikation ausgewählten Buckets (Quell- und Ziel-Buckets) muss die S3-Versionsverwaltung aktiviert sein und jeder Bucket muss sich in einer anderen AWS-Region befinden. Weitere Informationen zu S3 Versioning finden Sie unter [Verwenden der Versionsverwaltung in Amazon-S3-Buckets](#).

Wählen Sie unter Ziel-Buckets einen oder mehrere Buckets aus, in die Sie Objekte replizieren möchten.

ℹ Note

Stellen Sie sicher, dass Sie über die erforderlichen Lese- und Replikationsberechtigungen verfügen, um die Replikation einzurichten. Andernfalls treten Fehler auf. Weitere Informationen finden Sie unter [Erstellen einer IAM-Rolle](#).

8. Wählen Sie im Abschnitt Replication rule configuration (Konfiguration der Replikationsregel) aus, ob die Replikationsregel bei ihrer Erstellung Enabled (Aktiviert) oder Disabled (Deaktiviert) sein soll.

ℹ Note

Sie können keinen Namen in das Feld Replication rule name (Name der Replikationsregel) eingeben. Die Namen der Replikationsregeln werden basierend auf Ihrer Konfiguration generiert, wenn Sie die Replikationsregel erstellen.

9. Wählen Sie im Abschnitt Scope (Bereich) den entsprechenden Bereich für Ihre Replikation aus.

- Um den gesamten Bucket zu replizieren, wählen Sie Apply to all objects in the bucket (Auf alle Objekte im Bucket anwenden).
- Um eine Teilmenge der Objekte im Bucket zu replizieren, wählen Sie Limit the scope of this rule using one or more filters (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken) aus.

Sie können Ihre Objekte nach Präfix, Objekt-Tags oder einer Kombination aus beiden filtern.

- Wenn Sie die Replikation auf alle Objekte beschränken möchten, die mit derselben Zeichenfolge beginnen (z. B. pictures), geben Sie ein Präfix in das Feld Prefix (Präfix) ein.

Wenn Sie ein Präfix eingeben, bei dem es sich um den Namen eines Ordners handelt, müssen Sie ein Trennzeichen wie z. B. / (Schrägstrich) verwenden, um die Hierarchieebene des Ordners anzugeben (z. B. pictures/). Weitere Informationen zu Präfixen finden Sie unter [Organisieren von Objekten mit Präfixen](#).

- Um alle Objekte mit einem oder mehreren Objekt-Tags zu replizieren, wählen Sie Add tag (Tag hinzufügen) aus und geben Sie das Schlüssel-Wert-Paar in die Felder ein. Wiederholen Sie den Vorgang, um ein weiteres Tag hinzuzufügen. Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

10. Scrollen Sie nach unten zum Abschnitt Additional replication options (Zusätzliche Replikationsoptionen) und wählen Sie die Replikationsoptionen aus, die Sie anwenden möchten.

Note

Wir empfehlen Ihnen, folgende Optionen anzuwenden:

- Replication time control (RTC) Begrenzung der Replikationszeit (RTC) – Um Ihre Daten innerhalb eines vorhersehbaren Zeitraums über verschiedene Regionen hinweg zu replizieren, können Sie die Begrenzung der S3-Replikationszeit (S3 RTC) verwenden. S3 RTC repliziert 99,99 Prozent der neuen in Amazon S3 gespeicherten Objekte innerhalb von 15 Minuten (gestützt auf ein Service Level Agreement). Weitere Informationen finden Sie unter [the section called “Verwenden der S3-Replikationszeitkontrolle”](#).

- Replication metrics and notifications (Replikationsmetriken und -benachrichtigungen)
 - Aktivieren Sie Amazon-CloudWatch-Metriken, um Replikationsereignisse zu überwachen.
- Replikation der Löschkennzeichnung – Durch S3-Löschvorgänge erstellte Löschkennzeichnungen werden repliziert. Durch Lebenszyklusregeln erstellte Löschkennzeichnungen werden nicht repliziert. Weitere Informationen finden Sie unter [Replizieren von Löschkennzeichnungen auf Buckets](#).

Es fallen zusätzliche Gebühren für S3 RTC und CloudWatch-Replikationsmetriken und -benachrichtigungen an. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#) und [Amazon CloudWatch – Preise](#).

11. Wenn Sie eine neue Replikationsregel schreiben, die eine bestehende ersetzt, wählen Sie I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Ich bestätige, dass diese vorhandenen Replikationsregeln durch Auswählen von „Replikationsregeln erstellen“ überschrieben werden).
12. Wählen Sie Replikationsregeln erstellen aus, um Ihre neue unidirektionale Replikationsregel zu erstellen und zu speichern.

Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point

Replikationsregeln ermöglichen ein automatisches und asynchrones Kopieren von Objekten über Buckets hinweg. Eine bidirektionale Replikationsregel stellt sicher, dass Daten zwischen zwei oder mehr Buckets in verschiedenen AWS-Regionen vollständig synchronisiert werden. Wenn die bidirektionale Replikation eingerichtet ist, wird eine Replikationsregel vom Quell-Bucket (DOC-EXAMPLE-BUCKET-1) zu dem Bucket, der die Replikate enthält (DOC-EXAMPLE-BUCKET-2) erstellt. Anschließend wird eine zweite Replikationsregel von dem Bucket, der die Replikate enthält (DOC-EXAMPLE-BUCKET-2), zum Quell-Bucket (DOC-EXAMPLE-BUCKET-1) erstellt.

Wie alle Replikationsregeln können Sie die bidirektionale Replikationsregel entweder auf den gesamten Amazon-S3-Bucket oder auf eine anhand eines Präfixes oder von Objekt-Tags gefilterte Teilmenge von Objekten anwenden. Sie können auch Metadatenänderungen an Ihren Objekten u synchronisieren, indem Sie [die Synchronisierung von Replikatänderungen](#) für jede Replikationsregel aktivieren. Sie können die Synchronisierung von Replikatänderungen über die Amazon-S3-Konsole, die AWS CLI, die AWS-SDKs, die Amazon-S3-REST-API oder AWS CloudFormation aktivieren.

Um den Replikationsfortschritt von Objekten und Objektmetadaten in Amazon CloudWatch zu überwachen, aktivieren Sie S3-Replikationsmetriken und -benachrichtigungen. Weitere Informationen finden Sie unter [Überwachen des Fortschritts mit Replikationsmetriken und Amazon-S3-Ereignisbenachrichtigungen](#).

So erstellen Sie eine bidirektionale Replikationsregel für Ihren Multi-Region Access Point

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Point aus, den Sie aktualisieren möchten.
4. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
5. Scrollen Sie nach unten zum Abschnitt Replication rules (Replikationsregeln) und wählen Sie dann Create replication rules (Replikationsregeln erstellen) aus.
6. Wählen Sie auf der Seite Create replication rules (Replikationsregeln erstellen) die Vorlage Replicate objects among all specified buckets (Replizieren von Objekten zwischen allen angegebenen Buckets) aus. Die Vorlage Replicate objects among all specified buckets (Replizieren von Objekten zwischen allen angegebenen Buckets) richtet die bidirektionale Replikation (mit Failover-Funktionen) für Ihre Buckets ein.

 **Important**

Wenn Sie Replikationsregeln mithilfe dieser Vorlage erstellen, ersetzen diese alle vorhandenen Replikationsregeln, die dem Bucket bereits zugewiesen sind.

Wenn Sie Replikationsregeln hinzufügen oder bestehende Regeln ändern möchten, anstatt sie zu ersetzen, wechseln Sie in der Konsole zur Registerkarte Management (Verwaltung) der einzelnen Buckets und bearbeiten Sie dann die Regeln im Abschnitt Replication rules (Replikationsregeln). Sie können bestehende Replikationsregeln auch über die AWS CLI, AWS-SDKs oder die Amazon-S3-REST-API hinzufügen oder ändern. Weitere Informationen finden Sie unter [Replikationskonfiguration](#).

7. Wählen Sie im Abschnitt Buckets mindestens zwei Buckets aus, aus denen Sie Objekte replizieren möchten. Für alle für die Replikation ausgewählten Buckets muss S3 Versioning aktiviert sein und jeder Bucket muss sich in einer anderen AWS-Region befinden. Weitere Informationen zu S3 Versioning finden Sie unter [Verwenden der Versionsverwaltung in Amazon-S3-Buckets](#).

Note

Stellen Sie sicher, dass Sie über die erforderlichen Lese- und Replikationsberechtigungen verfügen, um die Replikation einzurichten. Andernfalls treten Fehler auf. Weitere Informationen finden Sie unter [Erstellen einer IAM-Rolle](#).

8. Wählen Sie im Abschnitt Replication rule configuration (Konfiguration der Replikationsregel) aus, ob die Replikationsregel bei ihrer Erstellung Enabled (Aktiviert) oder Disabled (Deaktiviert) sein soll.

Note

Sie können keinen Namen in das Feld Replication rule name (Name der Replikationsregel) eingeben. Die Namen der Replikationsregeln werden basierend auf Ihrer Konfiguration generiert, wenn Sie die Replikationsregel erstellen.

9. Wählen Sie im Abschnitt Scope (Bereich) den entsprechenden Bereich für Ihre Replikation aus.
 - Um den gesamten Bucket zu replizieren, wählen Sie Apply to all objects in the bucket (Auf alle Objekte im Bucket anwenden).
 - Um eine Teilmenge der Objekte im Bucket zu replizieren, wählen Sie Limit the scope of this rule using one or more filters (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken) aus.


Sie können Ihre Objekte nach Präfix, Objekt-Tags oder einer Kombination aus beiden filtern.

- Wenn Sie die Replikation auf alle Objekte beschränken möchten, die mit derselben Zeichenfolge beginnen (z. B. pictures), geben Sie ein Präfix in das Feld Prefix (Präfix) ein.

Wenn Sie ein Präfix eingeben, das den Namen eines Ordners darstellt, müssen Sie / (Schrägstrich) als letztes Zeichen eingeben (z. B. pictures/).

- Um alle Objekte mit einem oder mehreren Objekt-Tags zu replizieren, wählen Sie Add tag (Tag hinzufügen) aus und geben Sie das Schlüssel-Wert-Paar in die Felder ein. Wiederholen Sie den Vorgang, um ein weiteres Tag hinzuzufügen. Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

10. Scrollen Sie nach unten zum Abschnitt Additional replication options (Zusätzliche Replikationsoptionen) und wählen Sie die Replikationsoptionen aus, die Sie anwenden möchten.

 Note

Wir empfehlen Ihnen, die folgenden Optionen anzuwenden, insbesondere wenn Sie Ihren Multi-Region Access Point so konfigurieren möchten, dass er Failover unterstützt:

- Replication time control (RTC) Begrenzung der Replikationszeit (RTC) – Um Ihre Daten innerhalb eines vorhersehbaren Zeitraums über verschiedene Regionen hinweg zu replizieren, können Sie die Begrenzung der S3-Replikationszeit (S3 RTC) verwenden. S3 RTC repliziert 99,99 Prozent der neuen in Amazon S3 gespeicherten Objekte innerhalb von 15 Minuten (gestützt auf ein Service Level Agreement). Weitere Informationen finden Sie unter [the section called “Verwenden der S3-Replikationszeitkontrolle”](#).
- Replication metrics and notifications (Replikationsmetriken und -benachrichtigungen) – Aktivieren Sie Amazon-CloudWatch-Metriken, um Replikationsereignisse zu überwachen.
- Replikation der Löschmarkierung – Durch S3-Löschvorgänge erstellte Löschmarkierungen werden repliziert. Durch Lebenszyklusregeln erstellte Löschmarkierungen werden nicht repliziert. Weitere Informationen finden Sie unter [Replizieren von Löschmarkierungen auf Buckets](#).
- Replica modification sync (Synchronisierung von Replikatänderungen) – Aktivieren Sie die Synchronisierung von Replikatänderungen für jede Replikationsregel, um auch Metadatenänderungen an Ihren Objekten zu synchronisieren. Weitere Informationen finden Sie unter [Aktivieren der Synchronisierung von Replikatänderungen](#).

Es fallen zusätzliche Gebühren für S3 RTC und CloudWatch-Replikationsmetriken und -benachrichtigungen an. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#) und [Amazon CloudWatch – Preise](#).

11. Wenn Sie eine neue Replikationsregel schreiben, die eine bestehende ersetzt, wählen Sie I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Ich bestätige, dass diese vorhandenen Replikationsregeln durch Auswählen von „Replikationsregeln erstellen“ überschrieben werden).

12. Wählen Sie **Create replication rules (Replikationsregeln erstellen)**, um Ihre neuen bidirektionalen Replikationsregeln zu erstellen und zu speichern.

Anzeigen der Replikationsregeln für Ihren Multi-Region Access Point

Mit Multi-Region Access Points können Sie entweder unidirektionale oder bidirektionale Replikationsregeln einrichten. Informationen zur Verwaltung Ihrer Replikationsregeln finden Sie unter [Verwalten von Replikationsregeln mit der Amazon-S3-Konsole](#).

So zeigen Sie die Replikationsregeln für Ihren Multi-Region Access Point an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich **Multi-Region Access Points** aus.
3. Wählen Sie den Namen Ihres Multi-Region Access Point aus.
4. Wählen Sie die Registerkarte **Replication and failover (Replikation und Failover)** aus.
5. Scrollen Sie nach unten bis zum Abschnitt **Replikationsregeln**. In diesem Abschnitt sind alle Replikationsregeln aufgeführt, die für Ihren Multi-Region Access Point erstellt wurden.

Note

Wenn Sie diesem Multi-Region Access Point einen Bucket von einem anderen Konto hinzugefügt haben, benötigen Sie die Berechtigung `s3:GetBucketReplication` vom Bucket-Eigentümer, um die Replikationsregeln für diesen Bucket anzeigen zu können.

Verwendung von Access Points mit mehreren Regionen mit unterstützten Vorgängen API

Amazon S3 unterstützt verschiedene Vorgänge, mit denen Sie Multi-Regions-Zugriffspunkte verwalten können. Amazon S3 verarbeitet einige dieser Vorgänge synchron und einige asynchron. Wenn Sie einen asynchronen Vorgang aufrufen, autorisiert Amazon S3 den angeforderten Vorgang zunächst synchron. Wenn die Autorisierung erfolgreich ist, gibt Amazon S3 ein Token zurück, mit dem Sie den Fortschritt und die Ergebnisse des angeforderten Vorgangs verfolgen können.

Note

Anforderungen, die über die Amazon-S3-Konsole erfolgen, sind immer synchron. Die Konsole wartet, bis die Anforderung abgeschlossen ist, bevor Sie eine weitere Anforderung senden können.

Sie können den aktuellen Status und die Ergebnisse asynchroner Operationen über die Konsole oder über AWS CLI AWS SDKs, oder einsehen. `DescribeMultiRegionAccessPointOperation` REST API Amazon S3 stellt ein Tracking-Token in der Antwort auf einen asynchronen Vorgang bereit. Sie schließen dieses Tracking-Token als Argument für `DescribeMultiRegionAccessPointOperation` ein. Wenn Sie das Tracking-Token einschließen, gibt Amazon S3 den aktuellen Status und die Ergebnisse des angegebenen Vorgangs zurück, einschließlich etwaiger Fehler oder relevanter Ressourceninformationen. Amazon S3 führt `DescribeMultiRegionAccessPointOperation`-Vorgänge synchron durch.

Alle Anforderungen auf Steuerebene zum Erstellen oder Verwalten von Multi-Region Access Points müssen an die Region US West (Oregon) weitergeleitet werden. Für Anforderungen von Multi-Region Access Point auf Steuerebene müssen keine Regionen angegeben werden. Für die Failover-Stuerebene von Multi-Region Access Points muss die Anforderung an eine der fünf unterstützten Regionen weitergeleitet werden. Weitere Informationen zu Regionen, die von Access Points in mehreren Regionen unterstützt werden, finden Sie unter [Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts](#)

Darüber hinaus müssen Sie dem Benutzer, der Rolle oder einer anderen AWS Identity and Access Management (IAM) Entität, die eine Anfrage zur Verwaltung eines Access Points mit mehreren Regionen stellt, die `s3:ListAllMyBuckets` Berechtigung erteilen.

Die folgenden Beispiele veranschaulichen, wie Multi-Region Access Points mit kompatiblen Operationen in Amazon S3 verwendet werden.

Themen

- [Kompatibilität von Access Points für mehrere Regionen mit und AWS-Services AWS SDKs](#)
- [Kompatibilität von Multi-Region Access Points mit S3-Operationen](#)
- [Anzeigen Ihrer Weiterleitungskonfiguration für Multi-Region Access Points](#)
- [Aktualisieren der zugrunde liegende Bucket-Richtlinie von Amazon S3](#)
- [Aktualisieren einer Weiterleitungskonfiguration von Multi-Region Access Points](#)

- [Hinzufügen eines Objekts zu einem Bucket in Ihrem Multi-Region Access Point](#)
- [Abrufen von Objekten von Ihrem Multi-Region Access Point](#)
- [Auflisten von Objekten, die in einem Bucket gespeichert sind, der Ihrem Multi-Region Access Point zugrunde liegt](#)
- [Verwenden Sie einen URL mit mehreren Regionen vorsignierten Access Points](#)
- [Verwenden eines Buckets, der mit „Zahlung durch den Anforderer“ konfiguriert ist, mit Multi-Region Access Points](#)

Kompatibilität von Access Points für mehrere Regionen mit und AWS-Services AWS SDKs


Um einen Multi-Region-Access Point mit Anwendungen zu verwenden, die einen Amazon S3 S3-Bucket-Namen benötigen, verwenden Sie den Amazon-Ressourcennamen (ARN) des Multi-Region-Access Points, wenn Sie Anfragen mit einem stellen. AWS SDK [Informationen darüber, welche mit Access Points in mehreren Regionen kompatibel AWS SDKs sind, finden Sie unter Kompatibilität mit AWS SDKs](#)

Kompatibilität von Multi-Region Access Points mit S3-Operationen

Sie können die folgenden Amazon S3 API S3-Datenebenenoperationen verwenden, um Aktionen an Objekten in Buckets durchzuführen, die Ihrem Multi-Region Access Point zugeordnet sind. Die folgenden S3-Operationen können Multi-Region-Access Points akzeptieren: ARNs

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjectTagging](#)
- [GetObject](#)
- [GetObjectAcl](#)
- [GetObjectLegalHold](#)
- [GetObjectRetention](#)
- [GetObjectTagging](#)

- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjectsV2](#)
- [ListParts](#)
- [PutObject](#)
- [PutObjectAcl](#)
- [PutObjectLegalHold](#)
- [PutObjectRetention](#)
- [PutObjectTagging](#)
- [RestoreObject](#)
- [UploadPart](#)

 Note

Multi-Region-Access Points unterstützen Kopiervorgänge, bei denen Multi-Region-Access Points nur als Ziel verwendet werden, wenn der Multi-Region-Access Point verwendet wird.
ARN

Sie können die folgenden Amazon-S3-Operationen auf Steuerebene verwenden, um Ihre Multi-Region Access Points zu erstellen und zu verwalten:

- [CreateMultiRegionAccessPoint](#)
- [DescribeMultiRegionAccessPointOperation](#)
- [GetMultiRegionAccessPoint](#)
- [GetMultiRegionAccessPointPolicy](#)
- [GetMultiRegionAccessPointPolicyStatus](#)
- [GetMultiRegionAccessPointRoutes](#)
- [ListMultiRegionAccessPoints](#)
- [PutMultiRegionAccessPointPolicy](#)
- [SubmitMultiRegionAccessPointRoutes](#)

Anzeigen Ihrer Weiterleitungskonfiguration für Multi-Region Access Points

AWS CLI

Mit dem folgenden Beispielbefehl wird Ihre Routenkonfiguration für Multi-Region Access Points abgerufen, sodass Sie die aktuellen Weiterleitungstatus für Ihre Buckets sehen können. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
```

SDK for Java

Im Folgenden wird SDK für Java-Code die Routenkonfiguration Ihres Access Points mit mehreren Regionen abgerufen, sodass Sie den aktuellen Routing-Status für Ihre Buckets einsehen können. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider)
    .build();

GetMultiRegionAccessPointRoutesRequest request =
    GetMultiRegionAccessPointRoutesRequest.builder()
        .accountId("111122223333")
        .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
        .build();

GetMultiRegionAccessPointRoutesResponse response =
    s3ControlClient.getMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

Der folgende JavaScript Code ruft Ihre Routenkonfiguration SDK für Multi-Region Access Point ab, sodass Sie den aktuellen Routing-Status für Ihre Buckets sehen können. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
const REGION = 'us-east-1'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new GetMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
      })
    )
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Der folgende SDK Python-Code ruft Ihre Multi-Region Access Point-Routenkonfiguration ab, sodass Sie den aktuellen Routing-Status für Ihre Buckets sehen können. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
s3.get_multi_region_access_point_routes(
    AccountId=111122223333,
    Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap)['Routes']
```

Aktualisieren der zugrunde liegende Bucket-Richtlinie von Amazon S3

Damit ein ordnungsgemäßer Zugriff gewährt wird, müssen Sie auch die zugrunde liegende Bucket-Richtlinie von Amazon S3 aktualisieren. In den folgenden Beispielen wird die Zugriffskontrolle an die Richtlinie für Multi-Region Access Points delegiert. Nachdem Sie die Zugriffskontrolle an die Richtlinie für Multi-Region Access Points delegiert haben, wird die Bucket-Richtlinie nicht mehr für

die Zugriffskontrolle verwendet, wenn Anforderungen über den Multi-Region Access Point gestellt werden.

Hier ist ein Beispiel für eine Bucket-Richtlinie, die die Zugriffskontrolle an die Richtlinie der Multi-Region Access Points delegiert. Wenn Sie diese Bucket-Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen. Um diese Richtlinie mithilfe des AWS CLI `put-bucket-policy` Befehls anzuwenden, wie im nächsten Beispiel gezeigt, speichern Sie die Richtlinie beispielsweise in einer Datei. `policy.json`

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": { "AWS": "*" },
    "Effect": "Allow",
    "Action": ["s3:*"],
    "Resource": ["arn:aws:s3:::111122223333/*", "arn:aws:s3:::amzn-s3-demo-bucket"],
    "Condition": {
      "StringEquals": {
        "s3:DataAccessPointAccount": "444455556666"
      }
    }
  }
}
```

Der folgende Beispielbefehl `put-bucket-policy` verknüpft die aktualisierte S3-Bucket-Richtlinie mit Ihrem S3-Bucket:

```
aws s3api put-bucket-policy
--bucket amzn-s3-demo-bucket
--policy file:///tmp/policy.json
```

Aktualisieren einer Weiterleitungskonfiguration von Multi-Region Access Points

Mit dem folgenden Beispielbefehl wird die Weiterleitungskonfiguration des Multi-Region Access Points aktualisiert. Weiterleitungsbefehle für den Multi-Region Access Point können in einer dieser fünf Regionen ausgeführt werden:

- `ap-southeast-2`
- `ap-northeast-1`
- `us-east-1`

- `us-west-2`
- `eu-west-1`

In einer Weiterleitungskonfiguration für den Multi-Region Access Point können Sie Buckets auf einen aktiven oder passiven Weiterleitungsstatus festlegen. Aktive Buckets empfangen Traffic, passive Buckets nicht. Sie können den Weiterleitungsstatus eines Buckets festlegen, indem Sie den Wert `TrafficDialPercentage` für den Bucket auf `100` für aktiv oder `0` für passiv einstellen.

AWS CLI

Mit dem folgenden Beispielbefehl wird die Weiterleitungskonfiguration Ihres Multi-Region Access Points aktualisiert. In diesem Beispiel ist `amzn-s3-demo-bucket1` auf den aktiven Status und `amzn-s3-demo-bucket2` auf den passiven Status eingestellt. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
--route-updates Bucket=amzn-s3-demo-bucket1,TrafficDialPercentage=100
                  Bucket=amzn-s3-demo-bucket2,TrafficDialPercentage=0
```

SDK for Java

Im Folgenden SDK für Java-Code wird Ihre Routing-Konfiguration für multiregionale Access Points aktualisiert. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.ap-southeast-2)
    .credentialsProvider(credentialsProvider)
    .build();

SubmitMultiRegionAccessPointRoutesRequest request =
    SubmitMultiRegionAccessPointRoutesRequest.builder()
        .accountId("111122223333")
        .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
        .routeUpdates(
            MultiRegionAccessPointRoute.builder()
                .region("eu-west-1")
                .trafficDialPercentage(100)
```

```
        .build(),
    MultiRegionAccessPointRoute.builder()
        .region("ca-central-1")
        .bucket("111122223333")
        .trafficDialPercentage(0)
        .build()
    )
    .build();
```

```
SubmitMultiRegionAccessPointRoutesResponse response =
    s3ControlClient.submitMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

Der folgende JavaScript Code aktualisiert Ihre Routenkonfiguration SDK für multiregionale Access Points. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
const REGION = 'ap-southeast-2'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new SubmitMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3:::111122223333:accesspoint/abcdef0123456.mrap',
        RouteUpdates: [
          {
            Region: 'eu-west-1',
            TrafficDialPercentage: 100,
          },
          {
            Region: 'ca-central-1',
            Bucket: 'amzn-s3-demo-bucket1',
            TrafficDialPercentage: 0,
          },
        ],
      })
    )
  }
}
```

```
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Der folgende Python-Code aktualisiert Ihre Routenkonfiguration SDK für Multi-Region Access Point. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
s3.submit_multi_region_access_point_routes(
    AccountId=111122223333,
    Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrp,
    RouteUpdates= [{
        'Bucket': DOC-EXAMPLE-BUCKET,
        'Region': ap-southeast-2,
        'TrafficDialPercentage': 10
    }])
```

Hinzufügen eines Objekts zu einem Bucket in Ihrem Multi-Region Access Point

Verwenden Sie die Operation [PutObject](#), um ein Objekt zu dem Bucket hinzuzufügen, der dem Multi-Region Access Point zugeordnet ist. Aktivieren Sie die [regionsübergreifende Replikation](#), um alle Buckets im Multi-Region Access Point synchron zu halten.

Note

Für diese Operation müssen Sie über die `s3:PutObject`-Berechtigung für den Multi-Region Access Point verfügen. Weitere Informationen zu den Berechtigungsanforderungen für Multi-Region Access Points finden Sie unter [Berechtigungen](#).

AWS CLI

Das folgende Beispiel für Uploads von Anfragen auf Datenebene *example.txt* zum angegebenen multiregionalen Access Point. Um dieses Beispiel zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3api put-object --bucket
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --key example.txt --
body example.txt
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
    .build();

PutObjectRequest objectRequest = PutObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example.txt")
    .build();

s3Client.putObject(objectRequest, RequestBody.fromString("Hello S3!"));
```

SDK for JavaScript

```
const client = new S3Client({});

async function putObjectExample() {
  const command = new PutObjectCommand({
    Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
    Key: "example.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.put_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt',
    Body='Hello S3!'
)
```

Abrufen von Objekten von Ihrem Multi-Region Access Point

Verwenden Sie die Operation [GetObject](#), um Objekte aus dem Multi-Region Access Point abzurufen.

Note

Um diesen API Vorgang verwenden zu können, benötigen Sie die `s3:GetObject` Genehmigung für den Multi-Region Access Point. Weitere Informationen zu den Berechtigungsanforderungen für Multi-Region Access Points finden Sie unter [Berechtigungen](#).

AWS CLI

Im folgenden Beispiel wird eine Datenebenenanforderung abgerufen *example.txt* vom angegebenen Multi-Region-Access Point und lädt ihn herunter als *downloaded_example.txt*. Um dieses Beispiel zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3api get-object --bucket
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --
key example.txt downloaded_example.txt
```

SDK for Java

```
S3Client s3 = S3Client
    .builder()
    .build();
```

```
GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example.txt")
    .build();

s3Client.getObject(getObjectRequest);
```

SDK for JavaScript

```
const client = new S3Client({})

async function getObjectExample() {
  const command = new GetObjectCommand({
    Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
    Key: "example.txt"
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.get_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt'
)
```

Auflisten von Objekten, die in einem Bucket gespeichert sind, der Ihrem Multi-Region Access Point zugrunde liegt

Verwenden Sie die Operation [ListObjectsV2](#), um eine Liste von Objekten zurückzugeben, die in einem Bucket gespeichert sind, der Ihrem Multi-Region Access Point zugrunde liegt. Im folgenden

Beispielfehl werden alle Objekte für den angegebenen Multi-Region-Access Point mithilfe von ARN für den Multi-Region-Access Point aufgelistet. In diesem Fall ist der Multi-Region-Zugriffspunkt: ARN

```
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap
```

Note

Um diesen API Vorgang verwenden zu können, benötigen Sie die `s3:ListBucket` Berechtigung für den Multi-Region Access Point und den zugrunde liegenden Bucket. Weitere Informationen zu den Berechtigungsanforderungen für Multi-Region Access Points finden Sie unter [Berechtigungen](#).

AWS CLI

Das folgende Beispiel für eine Datenebenen-anforderung listet die Objekte in dem Bucket auf, der dem Multi-Region Access Point zugrunde liegt, der durch den spezifiziert wurde. ARN Um dieses Beispiel zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3api list-objects-v2 --bucket
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
    .build();

String bucketName = "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap";

ListObjectsV2Request listObjectsRequest = ListObjectsV2Request
    .builder()
    .bucket(bucketName)
    .build();

s3Client.listObjectsV2(listObjectsRequest);
```

SDK for JavaScript

```
const client = new S3Client({});
```

```
async function listObjectsExample() {
  const command = new ListObjectsV2Command({
    Bucket: "arn:aws:s3:::123456789012:accesspoint/abcdef0123456.mrap",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.list_objects_v2(
    Bucket='arn:aws:s3:::123456789012:accesspoint/abcdef0123456.mrap'
)
```

Verwenden Sie einen URL mit mehreren Regionen vorsignierten Access Points

Sie können einen vorsignierten verwenden, um einen URL zu generieren, der es anderen ermöglicht URL, über einen Amazon S3 S3-Zugriffspunkt für mehrere Regionen auf Ihre Amazon S3 S3-Buckets zuzugreifen. Wenn Sie ein vorsigniertes Objekt erstellen URL, verknüpfen Sie es mit einer bestimmten Objektaktion, z. B. einem S3-Upload (PutObject) oder einem S3-Download (GetObject). Sie können das vorsignierte Objekt teilen URL, und jeder, der Zugriff darauf hat, kann die darin eingebettete Aktion so ausführen, URL als wäre es der ursprüngliche signierende Benutzer.

Vorsignierte URLs haben ein Ablaufdatum. Wenn die Ablaufzeit erreicht ist, URL funktioniert das nicht mehr.

Bevor Sie S3 Multi-Region Access Points mit vorsignierter Version verwenden URLs, überprüfen Sie die [AWS SDK Kompatibilität](#) mit dem SigV4A-Algorithmus. Stellen Sie sicher, dass Ihre SDK Version SigV4a als Signaturimplementierung unterstützt, die zum Signieren der globalen Anfragen verwendet wird. AWS-Region Weitere Informationen zur Verwendung von presigned URLs with Amazon S3 finden Sie unter [Sharing objects by using URLs presigned](#).

Die folgenden Beispiele zeigen, wie Sie multiregionale Access Points mit vorsignierten Geräten verwenden können. URLs Um diese Beispiele zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

AWS CLI

```
aws s3 presign
arn:aws:s3::123456789012:accesspoint/MultiRegionAccessPoint_alias/example-file.txt
```

SDK for Python

```
import logging
import boto3
from botocore.exceptions import ClientError

s3_client = boto3.client('s3',aws_access_key_id='xxx',aws_secret_access_key='xxx')
s3_client.generate_presigned_url(HttpMethod='PUT',ClientMethod="put_object",
    Params={'Bucket':'arn:aws:s3::123456789012:accesspoint/
abcdef0123456.mrap','Key':'example-file'})
```

SDK for Java

```
S3Presigner s3Presigner = S3Presigner.builder()
    .credentialsProvider(StsAssumeRoleCredentialsProvider.builder()
        .refreshRequest(assumeRole)
        .stsClient(stsClient)
        .build())
    .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example-file")
    .build();

GetObjectPresignRequest preSignedReq = GetObjectPresignRequest.builder()
    .getObjectRequest(getObjectRequest)
    .signatureDuration(Duration.ofMinutes(10))
    .build();

PresignedGetObjectRequest presignedGetObjectRequest =
    s3Presigner.presignGetObject(preSignedReq);
```

Note

Um SigV4a mit temporären Sicherheitsanmeldedaten zu verwenden — zum Beispiel bei der Verwendung von IAM Rollen — stellen Sie sicher, dass Sie die temporären Anmeldeinformationen von einem regionalen Endpunkt in AWS Security Token Service (AWS STS) und nicht von einem globalen Endpunkt anfordern. Wenn Sie den globalen Endpunkt für AWS STS (`sts.amazonaws.com`) verwenden, werden temporäre Anmeldeinformationen von einem globalen Endpunkt generiert, was von Sig4A nicht unterstützt wird. Aus diesem Grund erhalten Sie eine Fehlermeldung. Um dieses Problem zu beheben, verwenden Sie einen der aufgelisteten [regionalen Endpunkte](#) für AWS STS

Verwenden eines Buckets, der mit „Zahlung durch den Anforderer“ konfiguriert ist, mit Multi-Region Access Points

Wenn ein S3-Bucket, der Ihren Multi-Region Access Points zugeordnet ist, [für die Verwendung von „Zahlung durch den Anforderer“ konfiguriert](#) ist, trägt der Anforderer die Kosten für die Bucket-Anforderung, den Download und alle mit Multi-Region Access Points verbundenen Kosten. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Hier sehen Sie ein Beispiel für eine Anforderung auf Datenebene an einen Multi-Region Access Point, der mit einem Bucket mit Zahlung durch den Anforderer verbunden ist.

AWS CLI

Wenn Sie Objekte von einem Multi-Region Access Point herunterladen möchten, der mit einem Bucket mit Zahlung durch den Anforderer verbunden ist, müssen Sie `--request-payer requester` als Teil Ihrer Anforderung [get-object](#) angeben. Sie müssen auch den Namen der Datei im Bucket sowie den Ort angeben, an dem sie gespeichert werden soll.

```
aws s3api get-object --bucket MultiRegionAccessPoint_ARN --request-payer requester  
--key example-file-in-bucket.txt example-location-of-downloaded-file.txt
```

SDK for Java

Wenn Sie Objekte von einem Multi-Region Access Point herunterladen möchten, der mit einem Bucket mit Zahlung durch den Anforderer verbunden ist, müssen Sie `RequestPayer.REQUESTER` als Teil Ihrer Anforderung `GetObject` angeben. Sie müssen auch den Namen der Datei in dem Bucket sowie den Ort angeben, an dem sie gespeichert werden soll.

```
GetObjectResponse getObjectResponse = s3Client.getObject(GetObjectRequest.builder()
    .key("example-file.txt")
    .bucket("arn:aws:s3::
123456789012:accesspoint/abcdef0123456.mrap")
    .requestPayer(RequestPayer.REQUESTER)
    .build()
).response();
```

Überwachung und Protokollierung von Anforderungen, die über einen Multi-Regions-Zugriffspunkt an zugrunde liegende Ressourcen erfolgen

Amazon S3 protokolliert Anforderungen, die über Multi-Region Access Points gestellt werden, und Anforderungen an die API-Operationen, die sie verwalten, wie z. B. `CreateMultiRegionAccessPoint` und `GetMultiRegionAccessPointPolicy`. Anforderungen, die über einen Multi-Regions-Zugriffspunkt an Amazon S3 gesendet werden, werden in den Zugriffsprotokollen und AWS CloudTrail-Protokollen des Amazon S3-Servers mit dem Hostnamen des Multi-Regions-Zugriffspunkts angezeigt. Der Hostname eines Zugriffspunkts hat das Format `MRAP_alias.accesspoint.s3-global.amazonaws.com`. Angenommen, Sie haben die folgende Bucket- und Multi-Regions-Zugriffspunktconfiguration:

- Ein Bucket mit dem Namen `my-bucket-usw2` in der Region `us-west-2`, der das Objekt `my-image.jpg` enthält.
- Ein Bucket mit dem Namen `my-bucket-aps1` in der Region `ap-south-1`, der das Objekt `my-image.jpg` enthält.
- Ein Bucket mit dem Namen `my-bucket-euc1` in der Region `eu-central-1`, der kein Objekt namens `my-image.jpg` enthält.
- Ein Multi-Regions-Zugriffspunkt namens `my-mrap` mit dem Alias `mfzwi23gnjvgw.mrap`, der so konfiguriert ist, dass Anforderungen aus allen drei Buckets erfüllt werden.
- Ihre AWS-Konto-ID ist `123456789012`.

Eine Anforderung, die gesendet wird, um `my-image.jpg` direkt über einen der Buckets anzurufen, wird in Ihren Protokollen mit dem Hostnamen `bucket_name.s3.Region.amazonaws.com` angezeigt.

Wenn Sie die Anforderung stattdessen über den Multi-Region Access Point stellen, bestimmt Amazon S3 zunächst, welcher der Buckets in den verschiedenen Regionen am nächsten gelegen ist.

Nach der Ermittlung, welcher Bucket zur Erfüllung der Anforderung verwendet werden soll, sendet Amazon S3 die Anforderung an diesen Bucket und protokolliert den Vorgang unter Verwendung des Hostnamens des Multi-Region Access Point. Wenn Amazon S3 in diesem Beispiel die Anforderung an `my-bucket-aps1` weiterleitet, spiegeln Ihre Protokolle eine erfolgreiche GET-Anforderung für `my-image.jpg` von `my-bucket-aps1` unter Verwendung des Hostnamens `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com` wider.

Important

Multi-Region Access Points kennen den Dateninhalt der zugrunde liegenden Buckets nicht. Daher kann es sein, dass der Bucket, der die Anforderung erhält, die angeforderten Daten nicht enthält. Wenn Amazon S3 beispielsweise feststellt, dass der Bucket `my-bucket-euc1` am nächsten gelegen ist, spiegeln Ihre Protokolle eine fehlgeschlagene GET-Anforderung für `my-image.jpg` von `my-bucket-euc1` unter Verwendung des Hostnamens `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com` wider. Wenn die Anforderung stattdessen an `my-bucket-usw2` weitergeleitet worden wäre, würden Ihre Protokolle eine erfolgreiche GET-Anforderung anzeigen.

Weitere Informationen zu Amazon S3-Server-Zugriffsprotokollen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#). Weitere Informationen zu AWS CloudTrail, finden Sie unter [Was ist AWS CloudTrail?](#) im AWS CloudTrail-Benutzerhandbuch.

Überwachen und Protokollieren von Anforderungen an Verwaltungs-API-Operationen von Multi-Region Access Points

Amazon S3 bietet verschiedene API-Operationen für die Verwaltung von Multi-Region Access Points, wie z. B. `CreateMultiRegionAccessPoint` und `GetMultiRegionAccessPointPolicy`. Wenn Sie Anforderungen an diese API-Operationen über die AWS Command Line Interface (AWS CLI), AWS-SDKs oder die Amazon-S3-REST-API stellen, verarbeitet Amazon S3 diese Anforderungen asynchron. Vorausgesetzt, Sie verfügen über die entsprechenden Berechtigungen für die Anforderung, gibt Amazon S3 ein Token für diese Anforderungen zurück. Sie können dieses Token mit `DescribeAsyncOperation` verwenden, um Ihnen zu helfen, den Status von laufenden asynchronen Vorgängen anzuzeigen. Amazon S3 verarbeitet `DescribeAsyncOperation`-Anforderungen synchron. Sie können die Amazon-S3-Konsole, die AWS CLI, SDKs oder die REST-API verwenden, um den Status asynchroner Anforderungen anzuzeigen.

 Note

Die Konsole zeigt nur den Status asynchroner Anforderungen an, die innerhalb der letzten 14 Tage gestellt wurden. Verwenden Sie AWS CLI, SDKs oder REST-API, um den Status älterer Anforderungen anzuzeigen.

Asynchrone Verwaltungsvorgänge können einen von mehreren Zuständen aufweisen:

NEW

Amazon S3 hat die Anforderung erhalten und bereitet die Ausführung des Vorgangs vor.

IN_PROGRESS

Amazon S3 führt derzeit den Vorgang aus.

SUCCESS

Der Vorgang war erfolgreich. Die Antwort enthält relevante Informationen, z. B. den Multi-Regions-Zugriffspunkt-Alias für eine `CreateMultiRegionAccessPoint`-Anforderung.

FAILED

Der Vorgang schlägt fehl. Die Antwort enthält eine Fehlermeldung, die den Grund für den Anforderungsfehler angibt.

Verwenden von AWS CloudTrail mit Multi-Region Access Points

Mit AWS CloudTrail können Sie Kontoaktivitäten in Ihrer AWS-Infrastruktur anzeigen, suchen, herunterladen, archivieren, analysieren und auf diese reagieren. Mit Multi-Region Access Points und CloudTrail-Protokollierung können Sie Folgendes ermitteln:

- Wer oder was welche Maßnahme ergriffen hat
- Welche Ressourcen in Anspruch genommen wurden
- Wann das Ereignis aufgetreten ist
- Weitere Details zu dem Ereignis

Sie können diese Protokollinformationen für die Analyse und Reaktion auf Aktivitäten verwenden, die über Ihre Multi-Region Access Points erfolgt sind.

Einrichten von AWS CloudTrail-Multi-Regions-Zugriffspunkten

Um die CloudTrail-Protokollierung für Vorgänge im Zusammenhang mit dem Erstellen oder Verwalten von Multi-Region Access Points zu aktivieren, müssen Sie die CloudTrail-Protokollierung so konfigurieren, dass die Ereignisse in der Region USA West (Oregon) aufgezeichnet werden. Sie müssen Ihre Protokollierungskonfiguration auf diese Weise einrichten, unabhängig davon, in welcher Region Sie sich befinden, wenn Sie die Anforderung stellen, oder welche Regionen der Multi-Region Access Point unterstützt. Alle Anforderungen, einen Multi-Region Access Point zu erstellen oder zu verwalten, werden über die Region USA West (Oregon) geleitet. Wir empfehlen Ihnen, diese Region entweder einem vorhandenen Trail hinzuzufügen oder einen neuen Trail zu erstellen, der diese Region und alle Regionen enthält, die mit dem Multi-Region Access Point verknüpft sind.

Amazon S3 protokolliert alle Anforderungen, die über einen Multi-Region Access Point erfolgen, und Anforderungen an die API-Operationen, die Zugriffspunkte verwalten, z. B. `CreateMultiRegionAccessPoint` und `GetMultiRegionAccessPointPolicy`. Wenn Sie diese Anforderungen über einen Multi-Regions-Zugriffspunkt protokollieren, werden sie in Ihren AWS CloudTrail-Protokollen mit dem Hostnamen des Multi-Regions-Zugriffspunkts erscheinen. Wenn Sie beispielsweise Anforderungen an einen Bucket über einen Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap` stellen, haben die Einträge im CloudTrail-Protokoll den Hostnamen `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

Multi-Region Access Points leiten Anforderungen an den nächstgelegenen Bucket weiter. Aufgrund dieses Verhaltens sind beim Betrachten der CloudTrail-Protokolle für einen Multi-Region Access Point Anforderungen an die zugrunde liegenden Buckets zu sehen. Einige dieser Anforderungen sind möglicherweise direkte Anforderungen an den Bucket, die nicht über den Multi-Region Access Point geleitet werden. Beachten Sie dies bei der Überprüfung des Datenverkehrs. Wenn sich ein Bucket in einem Multi-Regions-Zugriffspunkt befindet, können Anforderungen direkt an diesen Bucket gestellt werden, ohne den Multi-Regions-Zugriffspunkt zu durchlaufen.

Beim Erstellen und Verwalten von Multi-Regions-Zugriffspunkten sind asynchrone Ereignisse erforderlich. Asynchrone Anforderungen haben keine Abschlussereignisse im CloudTrail-Protokoll. Weitere Informationen zu asynchronen Anforderungen finden Sie unter [Überwachen und Protokollieren von Anforderungen an Verwaltungs-API-Operationen von Multi-Region Access Points](#).

Weitere Informationen zu AWS CloudTrail, finden Sie unter [Was ist AWS CloudTrail?](#) im AWS CloudTrail-Benutzerhandbuch.

Amazon-S3-Sicherheit

Cloud-Sicherheit bei AWS hat höchste Priorität. Als AWS Als Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung zwischen AWS und du. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

Sicherheit der Cloud

AWS ist verantwortlich für den Schutz der Infrastruktur, die läuft AWS Dienstleistungen in der AWS Cloud. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Die Wirksamkeit unserer Sicherheit wird regelmäßig von externen Prüfern im Rahmen der [AWS Compliance-Programme](#). Weitere Informationen zu den Compliance-Programmen, die für Amazon S3 gelten, finden Sie unter [AWS Dienstleistungen im Geltungsbereich nach Compliance-Programmen](#).

Sicherheit in der Cloud

Ihre Verantwortung wird bestimmt durch AWS Dienst, den Sie nutzen. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften. Für Amazon S3 umfasst Ihre Verantwortlichkeit die folgenden Bereiche:

- Verwalten Ihrer Daten, einschließlich [Objekt-Eigentumsrechte](#) und [Verschlüsselung](#).
- Klassifizierung Ihres Vermögens.
- [Verwaltung des Zugriffs auf](#) Ihre Daten mithilfe von [IAMRollen](#) und anderen Dienstkonfigurationen, um die entsprechenden Berechtigungen anzuwenden.
- Aktivierung von detektivischen Kontrollen wie [AWS CloudTrail](#) oder [Amazon GuardDuty](#) für Amazon S3.

Diese Dokumentation beschreibt, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon S3 anwenden können. Die folgenden Themen veranschaulichen, wie Sie Amazon S3 zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie werden auch lernen, andere zu verwenden AWS Services, die Ihnen helfen können, Ihre Amazon S3 S3-Ressourcen zu überwachen und zu sichern.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Datenschutz in Amazon S3](#)
- [Datenschutz durch Verschlüsselung](#)
- [Richtlinie für den Datenverkehr zwischen Netzwerken](#)
- [AWS PrivateLink für Amazon S3](#)
- [Zugriffsverwaltung](#)
- [Protokollierung und Überwachung in Amazon S3](#)
- [Compliance-Validierung für Amazon S3](#)
- [Ausfallsicherheit bei Amazon S3](#)
- [Infrastruktursicherheit in Amazon S3](#)
- [Konfigurations- und Schwachstellenanalyse in Amazon S3](#)
- [Bewährte Methoden für die Sicherheit in Amazon S3](#)
- [Überwachung der Datensicherheit mit Managed AWS Sicherheitsdienste](#)

Datenschutz in Amazon S3

Amazon S3 bietet eine sehr robuste Speicherinfrastruktur, die für geschäftskritische und primäre Speicheranwendungen entwickelt wurde. S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive speichern Objekte redundant auf mehreren Geräten in mindestens drei Availability Zones in einem AWS-Region. Eine Availability Zone ist eines oder mehrere diskrete Rechenzentren mit redundanter Stromversorgung, Vernetzung und Konnektivität in einem . AWS-Region. Availability Zones sind physisch durch eine nennenswerte Entfernung von vielen Kilometern von allen anderen Availability Zones getrennt, obwohl sich alle innerhalb von 100 km (60 Meilen) voneinander befinden. Die Speicherklasse S3 One Zone-IA speichert Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone. Diese Services wurden entwickelt, um mit gleichzeitigen Geräteausfällen

umzugehen, indem sie verlorene Redundanz schnell erkennen und reparieren, und auch regelmäßig die Integrität Ihrer Daten mithilfe von Prüfsummen überprüfen.

Der Amazon-S3-Standard Speicher bietet folgende Funktionen:

- Unterstützung durch [Service Level Agreement von Amazon S3](#).
- Auf 99,999999999 %-ige Zuverlässigkeit und 99,99 %-ige Verfügbarkeit von Objekten über einen Zeitraum von einem Jahr ausgelegt.
- S3-Standard, S3 Intelligent-Tiering, S3-Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind alle darauf ausgelegt, Daten im Falle des Verlustes einer gesamten Amazon-S3-Availability-Zone zu erhalten.

Darüber hinaus schützt Amazon S3 Ihre Daten durch Versioning. Sie können Versioning verwenden, um sämtliche Versionen aller Objekte in Ihrem Amazon-S3-Bucket zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leicht wiederherstellen. Standardmäßig wird durch Anforderungen die zuletzt geschriebene Version abgerufen. Ältere Versionen eines Objekts können abgerufen werden, indem die entsprechende Version des Objekts in der Anforderung angegeben wird.

Neben S3 Versioning können Sie auch Amazon S3 Object Lock und S3 Replication verwenden, um Ihre Daten zu schützen. Weitere Informationen finden Sie im [Tutorial: Schutz von Daten in Amazon S3 vor versehentlichem Löschen oder Anwendungsfehlern mithilfe von S3 Versionierung, S3 Object Lock und S3 Replication](#).

Aus Datenschutzgründen empfehlen wir Ihnen, AWS-Konto Zugangsdaten und richten Sie individuelle Benutzerkonten ein mit AWS Identity and Access Management, sodass jeder Benutzer nur die Berechtigungen erhält, die zur Erfüllung seiner beruflichen Pflichten erforderlich sind.

Wenn Sie für den Zugriff FIPS 140-2 validierte kryptografische Module benötigen AWS über eine Befehlszeilenschnittstelle oder einen API, verwenden Sie einen Endpunkt. FIPS Weitere Informationen zu den verfügbaren FIPS Endpunkten finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Die folgenden bewährten Sicherheitsmethoden umfassen ebenfalls den Adressdatenschutz in Amazon S3:

- [Implement server-side encryption](#)
- [Enforce encryption of data in transit](#)

- [Consider using Macie with Amazon S3](#)
- [Identify and audit all your Amazon S3 buckets](#)
- [Monitor Amazon Web Services security advisories](#)

Datenschutz durch Verschlüsselung

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Dieser Datenschutz bezieht sich auf Daten bei der Übertragung (wenn sie zu Amazon S3 oder von diesem geschickt werden), ebenso wie auf ruhende Daten (die in Amazon-S3-Rechenzentren auf Datenträgern gespeichert sind). Sie können Daten während der Übertragung schützen, indem Sie Secure Socket Layer/Transport Layer Security (SSL/TLS) oder clientseitige Verschlüsselung verwenden. Sie haben folgende Optionen, um Daten im Ruhezustand in Amazon S3 zu schützen:

- Serverseitige Verschlüsselung — Amazon S3 verschlüsselt Ihre Objekte, bevor sie auf Festplatten gespeichert werden in AWS Rechenzentren und entschlüsselt die Objekte dann, wenn Sie sie herunterladen.

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder SSE - KMS verwenden. DSSE KMS

Weitere Informationen zu den einzelnen Optionen für die serverseitige Verschlüsselung finden Sie unter [Schützen von Daten mit serverseitiger Verschlüsselung](#).

Informationen zum Konfigurieren der serverseitigen Verschlüsselung finden Sie unter:

- [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln \(SSE-S3\) angeben](#)
 - [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#)
 - [the section called “Spezifizierung von DSSE - KMS”](#)
 - [Angaben der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln \(-C\) SSE](#)
- Clientseitige Verschlüsselung – Sie können Daten clientseitig verschlüsseln und die verschlüsselten Daten auf Amazon S3 hochladen. In diesem Fall verwalten Sie den Verschlüsselungsprozess, die Verschlüsselungsschlüssel und die zugehörigen Tools.

Informationen zum Konfigurieren der clientseitigen Verschlüsselung finden Sie unter [Schützen von Daten mithilfe der clientseitigen Verschlüsselung](#).

Wenn Sie feststellen möchten, welcher Prozentsatz Ihrer Speicherbytes verschlüsselt ist, können Sie die Metriken von Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter [Bewertung Ihrer Speicheraktivität und -nutzung mit S3 Storage Lens](#). Eine vollständige Liste der Metriken finden Sie im [Glossar der S3-Storage-Lens-Metriken](#).

Weitere Informationen zur serverseitigen Verschlüsselung und zur clientseitigen Verschlüsselung finden Sie in den unten aufgeführten Themen.

Themen

- [Schützen von Daten mit serverseitiger Verschlüsselung](#)

- [Schützen von Daten mithilfe der clientseitigen Verschlüsselung](#)

Schützen von Daten mit serverseitiger Verschlüsselung


Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Serverseitige Verschlüsselung ist die Verschlüsselung von Daten am Zielort durch die Anwendung oder den Service, der sie erhält. Amazon S3 verschlüsselt Ihre Daten auf Objektebene beim Schreiben auf Festplatten in AWS Rechenzentren und entschlüsselt sie für Sie, wenn Sie darauf zugreifen. Wenn Sie Ihre Anforderung authentifizieren und Zugriffsberechtigungen besitzen, gibt es in Bezug auf die Art und Weise, wie Sie auf verschlüsselte oder nicht verschlüsselte Objekte zugreifen, keinen Unterschied. Wenn Sie beispielsweise Ihre Objekte gemeinsam nutzen, indem Sie ein vorsigniertes Objekt verwenden, URL funktioniert das sowohl für verschlüsselte als auch für unverschlüsselte Objekte auf die gleiche Weise. Wenn Sie Objekte in Ihrem Bucket auflisten, geben die API Listenoperationen außerdem eine Liste aller Objekte zurück, unabhängig davon, ob sie verschlüsselt sind.

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder SSE - KMS verwenden. DSSE KMS

 Note

Sie können nicht gleichzeitig unterschiedliche Arten serverseitiger Verschlüsselung auf dasselbe Objekt anwenden.

Wenn Sie Ihre vorhandenen Objekte verschlüsseln müssen, verwenden Sie S3 Batch Operations und S3 Inventory. Weitere Informationen finden Sie unter [Verschlüsseln von Objekten mit Amazon S3 Batch Operations](#)) und [Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch..](#)

Sie haben vier sich gegenseitig ausschließende Optionen für die serverseitige Verschlüsselung, abhängig davon, wie Sie die Verschlüsselungsschlüssel verwalten und wie viele Verschlüsselungsebenen Sie anwenden möchten.

Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert. Die Standardoption für serverseitige Verschlüsselung ist mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3). Jedes Objekt wird mit einem eindeutigen Schlüssel verschlüsselt. Als zusätzliche Sicherheitsmaßnahme verschlüsselt SSE -S3 den Schlüssel selbst mit einem Stammschlüssel, den es regelmäßig wechselt. SSE-S3 verwendet eine der stärksten verfügbaren Blockchiffren, den 256-Bit-Advanced Encryption Standard (AES-256), um Ihre Daten zu verschlüsseln. Weitere Informationen finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS)

Serverseitige Verschlüsselung mit AWS KMS keys (SSE-KMS) wird durch eine Integration von bereitgestellt AWS KMS Service mit Amazon S3. Mit AWS KMS, Sie haben mehr Kontrolle über Ihre Schlüssel. Sie können beispielsweise separate Schlüssel anzeigen, Kontrollrichtlinien bearbeiten und den Schlüsseln in AWS CloudTrail. Darüber hinaus können Sie vom Kunden verwaltete Schlüssel erstellen und verwalten oder verwenden Von AWS verwaltete Schlüssel die für Sie, Ihren Service

und Ihre Region einzigartig sind. Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (-) DSSE KMS

Zweischichtige serverseitige Verschlüsselung mit AWS KMS keys (DSSE-KMS) ähnelt SSE - KMS, KMS wendet jedoch DSSE statt einer Ebene zwei einzelne Verschlüsselungsebenen auf Objektebene an. Da beide Verschlüsselungsebenen serverseitig auf ein Objekt angewendet werden, können Sie eine Vielzahl von AWS-Services und Tools zur Analyse von Daten in S3 unter Verwendung einer Verschlüsselungsmethode, die Ihren Compliance-Anforderungen gerecht wird. Weitere Informationen finden Sie unter [Verwendung einer zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüssel \(-\) DSSE KMS](#).

Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Bei serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verwalten Sie die Verschlüsselungsschlüssel, und Amazon S3 verwaltet die Verschlüsselung beim Schreiben auf Festplatten und die Entschlüsselung beim Zugriff auf Ihre Objekte. Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln \(-C\) SSE](#).

Amazon S3 verschlüsselt jetzt automatisch alle neuen Objekte

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. SSE-S3, das den 256-Bit-Advanced Encryption Standard (AES-256) verwendet, wird automatisch auf alle neuen Buckets und auf alle vorhandenen S3-Buckets angewendet, für die noch keine Standardverschlüsselung konfiguriert ist. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface (AWS CLI) und die AWS SDKs.

In den folgenden Abschnitten werden Fragen zu diesem Update beantwortet.

Ändert Amazon S3 die Standardverschlüsselungseinstellungen für meine vorhandenen Buckets, für die bereits die Standardverschlüsselung konfiguriert ist?

Nein. Es gibt keine Änderungen an der Standardverschlüsselungskonfiguration für einen vorhandenen Bucket, der bereits über SSE -S3 oder serverseitige Verschlüsselung mit verfügt AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) konfiguriert. Weitere Informationen zum Festlegen des Standardverschlüsselungsverhalten für Buckets finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#). Weitere Hinweise zu den SSE KMS Verschlüsselungseinstellungen -S3 und SSE - finden Sie unter [Schützen von Daten mit serverseitiger Verschlüsselung](#).

Wird die Standardverschlüsselung für meine vorhandenen Buckets aktiviert, für die keine Standardverschlüsselung konfiguriert ist?

Ja. Amazon S3 konfiguriert jetzt die Standardverschlüsselung für alle vorhandenen unverschlüsselten Buckets, um serverseitige Verschlüsselung mit verwalteten S3-Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für neue Objekte anzuwenden, die in diese Buckets hochgeladen werden. Objekte, die sich bereits in einem vorhandenen nicht verschlüsselten Bucket befinden, werden nicht automatisch verschlüsselt.

Wie kann ich den Standardverschlüsselungsstatus neuer Objekt-Uploads einsehen?

Derzeit können Sie den Standard-Verschlüsselungsstatus neuer Objekt-Uploads unter einsehen AWS CloudTrail Logs, S3 Inventory und S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface (AWS CLI) und die AWS SDKs.

- Informationen zum Anzeigen Ihrer CloudTrail Ereignisse finden Sie unter [CloudTrail Ereignisse in der CloudTrail Konsole anzeigen](#) im AWS CloudTrail Benutzerleitfaden. CloudTrail Protokolle ermöglichen die API Nachverfolgung von PUT und POST Anfragen an Amazon S3. Wenn die Standardverschlüsselung zur Verschlüsselung von Objekten in Ihren Buckets verwendet wird, enthalten die CloudTrail Protokolle für PUT und POST API Anfragen das folgende Feld als Name-Wert-Paar: "SSEApplied": "Default_SSE_S3"
- Um den automatischen Verschlüsselungsstatus neuer Objektuploads in S3 Inventar anzuzeigen, konfigurieren Sie einen S3-Inventarbericht, der das Feld Verschlüsselungsmetadaten enthält, und sehen Sie sich dann den Verschlüsselungsstatus jedes neuen Objekts im Bericht an. Weitere Informationen finden Sie unter [Einrichtung von Amazon S3 Inventory](#).
- Um den automatischen Verschlüsselungsstatus für neue Objektuploads in S3 Storage Lens anzuzeigen, konfigurieren Sie ein S3-Storage-Lens-Dashboard und sehen Sie sich die Metriken Verschlüsselte Bytes und Anzahl verschlüsselter Objekte in der Kategorie Datenschutz des

Dashboards an. Weitere Informationen erhalten Sie unter [Erstellen eines Amazon S3-Storage-Lens-Dashboards](#) und [Anzeigen von S3-Storage-Lens-Metriken in den Dashboards](#).

- Wenn Sie den Status der automatischen Verschlüsselung auf Bucket-Ebene in der Amazon-S3-Konsole anzeigen möchten, überprüfen Sie die Standardverschlüsselung Ihrer Amazon-S3-Buckets in der Amazon-S3-Konsole. Weitere Informationen finden Sie unter [Konfigurieren der Standardverschlüsselung](#).
- Um den Status der automatischen Verschlüsselung als zusätzlichen Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface (AWS CLI) und die AWS SDKs, überprüfen Sie den Antwort-Header APIs, `x-amz-server-side-encryption` wenn Sie Objektaktionen wie [PutObject](#) und verwenden [GetObject](#).

Was muss ich tun, um von dieser Änderung zu profitieren?

Sie müssen keine Änderungen an Ihren bestehenden Anwendungen vornehmen. Da die Standardverschlüsselung für alle Ihre Buckets aktiviert ist, werden alle neuen Objekte, die auf Amazon S3 hochgeladen werden, automatisch verschlüsselt.

Kann ich die Verschlüsselung für die neuen Objekte deaktivieren, die in meinen Bucket geschrieben werden?

Nein. SSE-S3 ist die neue Basisverschlüsselungsebene, die auf alle neuen Objekte angewendet wird, die in Ihren Bucket hochgeladen werden. Sie können die Verschlüsselung für das Hochladen neuer Objekte nicht mehr deaktivieren.

Werden sich meine Gebühren ändern?

Nein. Die Standardverschlüsselung mit SSE -S3 ist ohne zusätzliche Kosten verfügbar. Speicherplatz, Anfragen und andere S3-Funktionen werden Ihnen wie gewohnt in Rechnung gestellt. Preisinformationen finden Sie unter [Amazon S3 – Preise](#).

Verschlüsselt Amazon S3 meine vorhandenen unverschlüsselten Objekte?

Nein. Ab dem 5. Januar 2023 verschlüsselt Amazon S3 nur neue Objekt-Uploads automatisch. Zum Verschlüsseln bestehender Objekte können Sie S3-Batch-Operationen verwenden, um verschlüsselte Kopien Ihrer Objekte zu erstellen. Diese verschlüsselten Kopien behalten die vorhandenen Objektdaten und den Namen bei und werden mit den von Ihnen angegebenen Verschlüsselungsschlüsseln verschlüsselt. Weitere Informationen finden Sie unter [Verschlüsseln von Objekten mit Amazon S3 Batch Operations](#) in der AWS Speicher-Blog.

Ich habe vor dieser Version die Verschlüsselung für meine Buckets nicht aktiviert. Muss ich die Art und Weise, wie ich auf Objekte zugreife, ändern?

Nein. Die Standardverschlüsselung mit SSE -S3 verschlüsselt Ihre Daten automatisch, während sie in Amazon S3 geschrieben werden, und entschlüsselt sie für Sie, wenn Sie darauf zugreifen. Die Art und Weise, wie Sie auf automatisch verschlüsselte Objekte zugreifen, ändert sich nicht.

Muss ich die Art und Weise, wie ich auf meine clientseitig verschlüsselten Objekte zugreife, ändern?

Nein. Alle clientseitig verschlüsselten Objekte, die vor dem Hochladen in Amazon S3 verschlüsselt werden, kommen als verschlüsselte Geheimentextobjekte in Amazon S3 an. Diese Objekte verfügen jetzt über eine zusätzliche SSE -S3-Verschlüsselungsebene. Ihre Workloads, die clientseitig verschlüsselte Objekte verwenden, erfordern keine Änderungen an Ihren Client-Services oder Autorisierungseinstellungen.

Note

HashiCorp Terraform-Benutzer, die keine aktualisierte Version von verwenden AWS Nach der Erstellung neuer S3-Buckets ohne kundenspezifische Verschlüsselungskonfiguration kann es beim Anbieter zu einer unerwarteten Abweichung kommen. Um diese Abweichung zu vermeiden, aktualisieren Sie Ihre Terraform AWS Provider-Version auf eine der folgenden Versionen: beliebig 4.x Veröffentlichung, 3.76.1, oder 2.70.4.

Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3)

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Alle neuen Objekt-Uploads in Amazon S3 S3-Buckets werden standardmäßig mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt.

Die serverseitige Verschlüsselung schützt Daten im Ruhezustand. Amazon S3 verschlüsselt jedes Objekt mit einem eindeutigen Schlüssel. Als zusätzliche Sicherheit verschlüsselt es den Schlüssel selbst mit einem Schlüssel, der regelmäßig rotiert. Die serverseitige Amazon S3 S3-Verschlüsselung verwendet den 256-Bit-Advanced Encryption Standard Galois/Counter Mode (AES-GCM), um alle hochgeladenen Objekte zu verschlüsseln.

Für die Verwendung der serverseitigen Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) fallen keine zusätzlichen Gebühren an. Für Anfragen zum Konfigurieren der Standard-Verschlüsselungsfunktion werden jedoch Standardgebühren für Amazon-S3-Anfragen berechnet. Informationen zu Preisen finden Sie unter [Amazon S3 – Preise](#).

Wenn Sie möchten, dass Ihre Daten-Uploads ausschließlich mit von Amazon S3 verwalteten Schlüsseln verschlüsselt werden, können Sie die folgende Bucket-Richtlinie verwenden. Beispielsweise verweigert die folgende Bucket-Richtlinie Berechtigungen zum Hochladen von Objekten, wenn die Anforderung nicht den `x-amz-server-side-encryption`-Header enthält, der eine serverseitige Verschlüsselung anfordert:

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    }
  ]
}
```

Note

Die serverseitige Verschlüsselung verschlüsselt nur die Objektdaten, nicht die Metadaten des Objekts.

APIUnterstützung für serverseitige Verschlüsselung


Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder SSE - KMS verwenden. DSSE KMS

Um die serverseitige Verschlüsselung mithilfe der Objekterstellung zu konfigurieren REST APIs, müssen Sie den `x-amz-server-side-encryption` Anforderungsheader angeben. Hinweise zu dem finden Sie REST APIs unter [Unter Verwendung der REST API](#).

Die folgenden Amazon S3 APIs unterstützen diesen Header:

- PUT Operationen — Geben Sie den Anforderungsheader an, wenn Sie Daten mit dem PUT API hochladen. Weitere Informationen finden Sie unter [PUT Objekt](#).
- Mehrteiligen Upload initiieren — Geben Sie den Header in der Initialisierungsanforderung an, wenn Sie große Objekte mithilfe des mehrteiligen Upload-Vorgangs hochladen. API Weitere Informationen finden Sie unter [Mehrteiligen Upload initiieren](#).
- COPY Operationen — Wenn Sie ein Objekt kopieren, haben Sie sowohl ein Quell- als auch ein Zielobjekt. Weitere Informationen finden Sie unter [PUT Objekt — Kopieren](#).

 Note

Wenn Sie eine POST-Operation für das Hochladen eines Objekts verwenden, anstatt den Anforderungs-Header anzugeben, stellen Sie die gleichen Informationen in den Formularfeldern bereit. Weitere Informationen finden Sie unter [POSTObjekt](#).


Das Tool AWS SDKs stellen Sie auch einen Wrapper bereit APIs, mit dem Sie serverseitige Verschlüsselung anfordern können. Sie können auch den AWS Management Console um Objekte hochzuladen und serverseitige Verschlüsselung anzufordern.

Weitere allgemeine Informationen finden Sie unter [AWS KMS Konzepte](#) in der AWS Key Management Service Leitfaden für Entwickler.

Themen

- [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln \(SSE-S3\) angeben](#)

Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) angeben

 Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen

Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder SSE - KMS verwenden. DSSE KMS

Sie können SSE -S3 mithilfe der S3-Konsole angeben, REST APIs AWS SDKs, und AWS Command Line Interface (AWS CLI). Weitere Informationen finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

Verwenden der S3-Konsole

In diesem Thema wird beschrieben, wie Sie den Verschlüsselungstyp für ein Objekt mithilfe der AWS Management Console. Wenn Sie ein Objekt mithilfe der Konsole kopieren, kopiert Amazon S3 das Objekt unverändert. Wenn das Quellobjekt verschlüsselt ist, wird das Zielobjekt demnach ebenfalls verschlüsselt. Sie können mit der Konsole die Verschlüsselung für ein Objekt hinzufügen oder ändern.

Note

- Wenn Sie die Verschlüsselung eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).
- Wenn Sie den Verschlüsselungstyp für ein Objekt mit benutzerdefinierten Tags ändern, benötigen Sie die entsprechende `s3:GetObjectTagging` Genehmigung. Wenn Sie den Verschlüsselungstyp für ein Objekt ändern, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Genehmigung.

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, wird der Verschlüsselungstyp für das Objekt aktualisiert, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

So ändern Sie die Verschlüsselung für ein Objekt

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
4. Wählen Sie in der Liste Objekte den Namen des Objekts aus, für das Sie eine Verschlüsselung hinzufügen oder ändern möchten.

Die Detailseite des Objekts wird angezeigt. Sie enthält mehrere Abschnitte mit den Eigenschaften des Objekts.

5. Wählen Sie die Registerkarte Eigenschaften aus.
6. Scrollen Sie nach unten zum Abschnitt Serverseitige Verschlüsselungseinstellungen und wählen Sie dann Bearbeiten aus.
7. Wählen Sie unter Verschlüsselungseinstellungen die Option Verwenden von Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung aus.
8. Wenn Sie Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung ausgewählt haben, konfigurieren Sie die folgenden Verschlüsselungseinstellungen.
 - Wählen Sie unter Verschlüsselungstyp die Option Amazon S3 Managed Keys (SSE-S3) aus. SSE-S3 verwendet eine der stärksten Blockchiffren — den 256-Bit-Advanced Encryption Standard (-256), um jedes Objekt zu verschlüsseln. AES Weitere Informationen finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).
9. Wählen Sie Save Changes (Änderungen speichern).

Note

Diese Aktion wendet auf alle angegebenen Objekte Verschlüsselung an. Warten Sie beim Verschlüsseln von Ordnern, bis die Speicheroperation abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen.

Unter Verwendung der REST API

Zum Zeitpunkt der Objekterstellung, d. h. wenn Sie ein neues Objekt hochladen oder eine Kopie eines vorhandenen Objekts erstellen, können Sie angeben, ob Amazon S3 Ihre Daten mit Amazon S3-verwalteten Schlüsseln (SSE-S3) verschlüsseln soll, indem Sie der Anfrage den Header hinzufügen. `x-amz-server-side-encryption` Setzen Sie den Wert des Headers auf den Verschlüsselungsalgorithmus AES256, den Amazon S3 unterstützt. Amazon S3 bestätigt, dass Ihr Objekt mit SSE -S3 gespeichert ist, indem es den Antwort-Header `x-amz-server-side-encryption` zurückgibt.

Die folgenden REST API Upload-Operationen akzeptieren den `x-amz-server-side-encryption` Anforderungsheader.

- [PUTObjekt](#)
- [PUTObjekt — Kopieren](#)
- [POSTObjekt](#)
- [Initiieren eines mehrteiligen Uploads](#)

Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie die serverseitige Verschlüsselung angeben, indem Sie den `x-amz-server-side-encryption` Header zur Anforderung Multipart Upload initiieren hinzufügen. Beim Kopieren eines vorhandenen Objekts wird das Zielobjekt unabhängig davon, ob das Quellobjekt verschlüsselt ist, nur dann verschlüsselt, wenn Sie die serverseitige Verschlüsselung explizit anfordern.

Die Antwortheader der folgenden REST API Operationen geben den Header zurück, wenn ein Objekt mit `x-amz-server-side-encryption -S3` gespeichert wird. SSE

- [PUTObjekt](#)
- [PUTObjekt — Kopieren](#)
- [POSTObjekt](#)
- [Initiieren eines mehrteiligen Uploads](#)
- [Upload Part](#)
- [Hochladen eines Teiluploads – Kopieren](#)
- [Abschließen eines mehrteiligen Uploads](#)
- [Get Object](#)
- [Head Object](#)

Note

Senden Sie keine Header für GET Verschlüsselungsanforderungen für Anfragen und HEAD Anfragen, wenn Ihr Objekt SSE -S3 verwendet. Andernfalls erhalten Sie HTTP den Statuscode 400 (Bad Request).

Verwendung der AWS SDKs

Bei der Verwendung AWS SDKs, können Sie Amazon S3 auffordern, serverseitige Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) zu verwenden. Dieser Abschnitt enthält Beispiele für die Verwendung von AWS SDKs in mehreren Sprachen. Informationen zu anderen SDKs finden Sie unter [Beispielcode und Bibliotheken](#).

Java

Wenn Sie das verwenden AWS SDK for Java um ein Objekt hochzuladen, können Sie es mit SSE -S3 verschlüsseln. Um eine serverseitige Verschlüsselung anzufordern, legen Sie mit der `ObjectMetadata`-Eigenschaft der `PutObjectRequest` den Anforderungs-Header `x-amz-server-side-encryption` fest. Wenn Sie die Methode `putObject()` des `AmazonS3Client`-Clients aufrufen, verschlüsselt und speichert Amazon S3 die Daten.

Sie können auch die SSE -S3-Verschlüsselung anfordern, wenn Sie Objekte mit dem mehrteiligen Upload-Vorgang hochladen: API

- Wenn Sie den mehrteiligen API Upload-Vorgang auf hoher Ebene verwenden, verwenden Sie die `TransferManager` Methoden, um Objekte beim Hochladen serverseitig zu verschlüsseln. Sie können eine beliebige der Upload-Methoden nutzen, die `ObjectMetadata` als Parameter entgegennehmen. Weitere Informationen finden Sie unter [Hochladen eines Objekts mit Multipart-Upload](#).
- Wenn Sie den mehrteiligen API Upload-Vorgang auf niedriger Ebene verwenden, geben Sie die serverseitige Verschlüsselung an, wenn Sie den mehrteiligen Upload starten. Sie fügen die Eigenschaft `ObjectMetadata` beim Aufruf der Methode `InitiateMultipartUploadRequest.setObjectMetadata()` hinzu. Weitere Informationen finden Sie unter [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#).

Sie können den Verschlüsselungsstatus eines Objekts (Verschlüsseln eines unverschlüsselten Objekts oder Entschlüsseln eines verschlüsselten Objekts) nicht direkt ändern. Um den

Verschlüsselungsstatus eines Objekts zu ändern, erstellen Sie eine Kopie des Objekts, geben dabei den gewünschten Verschlüsselungsstatus der Kopie an und löschen dann das Originalobjekt. Amazon S3 verschlüsselt das kopierte Objekt nur, wenn Sie explizit eine serverseitige Verschlüsselung anfordern. Um die Verschlüsselung des kopierten Objekts über Java anzufordernAPI, verwenden Sie die `ObjectMetadata` Eigenschaft, um die serverseitige Verschlüsselung in der anzugeben. `CopyObjectRequest`

Example Beispiel

Das folgende Beispiel zeigt, wie Sie die serverseitige Verschlüsselung einrichten, indem Sie AWS SDK for Java. Es zeigt, wie die folgenden Aufgaben ausgeführt werden:

- Laden Sie mit SSE -S3 ein neues Objekt hoch.
- Ändern des Verschlüsselungsstatus eines Objekts (in diesem Beispiel Verschlüsseln eines zuvor unverschlüsselten Objekts) durch Anfertigen einer Kopie des Objekts
- Überprüfen des Verschlüsselungsstatus des Objekts

Weitere Informationen zur serverseitigen Verschlüsselung finden Sie unter [Unter Verwendung der REST API](#). Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.internal.SSEResultBase;
import com.amazonaws.services.s3.model.*;

import java.io.ByteArrayInputStream;

public class SpecifyServerSideEncryption {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyNameToEncrypt = "**** Key name for an object to upload and encrypt
****";
```

```
String keyNameToCopyAndEncrypt = "**** Key name for an unencrypted object to
be encrypted by copying ****";
String copiedObjectKeyName = "**** Key name for the encrypted copy of the
unencrypted object ****";

try {
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .withRegion(clientRegion)
        .withCredentials(new ProfileCredentialsProvider())
        .build();

    // Upload an object and encrypt it with SSE.
    uploadObjectWithSSEEncryption(s3Client, bucketName, keyNameToEncrypt);

    // Upload a new unencrypted object, then change its encryption state
    // to encrypted by making a copy.
    changeSSEEncryptionStatusByCopying(s3Client,
        bucketName,
        keyNameToCopyAndEncrypt,
        copiedObjectKeyName);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void uploadObjectWithSSEEncryption(AmazonS3 s3Client, String
bucketName, String keyName) {
    String objectContent = "Test object encrypted with SSE";
    byte[] objectBytes = objectContent.getBytes();

    // Specify server-side encryption.
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.setContentLength(objectBytes.length);

    objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
    PutObjectRequest putRequest = new PutObjectRequest(bucketName,
        keyName,
        new ByteArrayInputStream(objectBytes),
```

```

        objectMetadata);

    // Upload the object and check its encryption status.
    PutObjectResult putResult = s3Client.putObject(putRequest);
    System.out.println("Object \"" + keyName + "\" uploaded with SSE.");
    printEncryptionStatus(putResult);
}

private static void changeSSEEncryptionStatusByCopying(AmazonS3 s3Client,
    String bucketName,
    String sourceKey,
    String destKey) {
    // Upload a new, unencrypted object.
    PutObjectResult putResult = s3Client.putObject(bucketName, sourceKey,
"Object example to encrypt by copying");
    System.out.println("Unencrypted object \"" + sourceKey + "\" uploaded.");
    printEncryptionStatus(putResult);

    // Make a copy of the object and use server-side encryption when storing the
    // copy.
    CopyObjectRequest request = new CopyObjectRequest(bucketName,
        sourceKey,
        bucketName,
        destKey);
    ObjectMetadata objectMetadata = new ObjectMetadata();

    objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
    request.setNewObjectMetadata(objectMetadata);

    // Perform the copy operation and display the copy's encryption status.
    CopyObjectResult response = s3Client.copyObject(request);
    System.out.println("Object \"" + destKey + "\" uploaded with SSE.");
    printEncryptionStatus(response);

    // Delete the original, unencrypted object, leaving only the encrypted copy
in
    // Amazon S3.
    s3Client.deleteObject(bucketName, sourceKey);
    System.out.println("Unencrypted object \"" + sourceKey + "\" deleted.");
}

private static void printEncryptionStatus(SSEResultBase response) {
    String encryptionStatus = response.getSSEAlgorithm();
    if (encryptionStatus == null) {

```



```
        encryptionStatus = "Not encrypted with SSE";
    }
    System.out.println("Object encryption status is: " + encryptionStatus);
}
}
```

.NET

Wenn Sie ein Objekt hochladen, können Sie Amazon S3 dazu anweisen, es zu verschlüsseln. Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ändern, erstellen Sie eine Kopie des Objekts und löschen dann das Quellobjekt. Beachten Sie, dass die Kopieroperation das Ziel nur verschlüsselt, wenn Sie auf dem Zielobjekt ausdrücklich eine serverseitige Verschlüsselung anfordern. Um SSE -S3 in der anzugebenCopyObjectRequest, fügen Sie Folgendes hinzu:

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Ein funktionierendes Beispiel, das zeigt, wie ein Objekt kopiert wird, finden Sie unter [Verwendung der AWS SDKs](#).

Das folgende Beispiel lädt ein Objekt hoch. In der Anfrage weist das Beispiel Amazon S3 dazu an, das Objekt zu verschlüsseln. Das Beispiel ruft dann Objekt-Metadaten ab und überprüft die verwendete Verschlüsselungsmethode. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SpecifyServerSideEncryptionTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string keyName = "**** key name for object created ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
            RegionEndpoint.USWest2;
        private static IAmazonS3 client;
```

```
public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    WritingAnObjectAsync().Wait();
}

static async Task WritingAnObjectAsync()
{
    try
    {
        var putRequest = new PutObjectRequest
        {
            BucketName = bucketName,
            Key = keyName,
            ContentBody = "sample text",
            ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
        };

        var putResponse = await client.PutObjectAsync(putRequest);

        // Determine the encryption state of an object.
        GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
        {
            BucketName = bucketName,
            Key = keyName
        };
        GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);
        ServerSideEncryptionMethod objectEncryption =
response.ServerSideEncryptionMethod;

        Console.WriteLine("Encryption method used: {0}",
objectEncryption.ToString());
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
    }
    catch (Exception e)
    {

```

```
        Console.WriteLine("Unknown encountered on server. Message: '{0}' when  
writing an object", e.Message);  
    }  
}  
}
```

PHP

Dieses Thema zeigt, wie Sie Klassen aus Version 3 von verwenden AWS SDK for PHP um SSE - S3 zu Objekten hinzuzufügen, die Sie auf Amazon S3 hochladen. Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Verwenden Sie die Methode [Aws\S3\S3Client::putObject \(\)](#), um ein Objekt auf Amazon S3 hochzuladen. Um Ihrer Upload-Anfrage den Anfrage-Header x-amz-server-side-encryption hinzuzufügen, geben Sie den Parameter `ServerSideEncryption` mit dem Wert `AES256` an, wie im folgenden Codebeispiel veranschaulicht. Weitere Informationen zur serverseitigen Verschlüsselungsanforderungen finden Sie unter [Unter Verwendung der REST API](#).

```
require 'vendor/autoload.php';  
  
use Aws\S3\S3Client;  
  
$bucket = '*** Your Bucket Name ***';  
$keyname = '*** Your Object Key ***';  
  
// $filepath should be an absolute path to a file on disk.  
$filepath = '*** Your File Path ***';  
  
$s3 = new S3Client([  
    'version' => 'latest',  
    'region' => 'us-east-1'  
]);  
  
// Upload a file with server-side encryption.  
$result = $s3->putObject([  
    'Bucket' => $bucket,  
    'Key' => $keyname,  
    'SourceFile' => $filepath,  
    'ServerSideEncryption' => 'AES256',  
]);
```

Amazon S3 gibt als Antwort den `x-amz-server-side-encryption`-Header mit dem Wert zurück, den der Verschlüsselungsalgorithmus für die Verschlüsselung Ihrer Objektdaten verwendet hat.

Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie SSE -S3 für die Objekte, die Sie hochladen, wie folgt angeben:

- Wenn Sie den mehrteiligen API Upload-Vorgang auf niedriger Ebene verwenden, geben Sie beim Aufrufen der Methode [Aws\S3\S3Client::\(\)](#) die serverseitige Verschlüsselung an. `createMultipartUpload` Um Ihrer Upload-Anfrage den Anfrage-Header `x-amz-server-side-encryption` hinzuzufügen, geben Sie für den Parameter `array` den `ServerSideEncryption`-Schlüssel mit dem Wert `AES256` an. Weitere Hinweise zum mehrteiligen Upload-Vorgang auf niedriger Ebene finden Sie unter. API [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#)
- Wenn Sie den mehrteiligen API Upload-Vorgang auf hoher Ebene verwenden, geben Sie die serverseitige Verschlüsselung mithilfe des `ServerSideEncryption` Vorgangsparameters an. [CreateMultipartUpload](#)API Ein Beispiel für die Verwendung der `setOption()` Methode mit dem mehrteiligen API Upload-Vorgang auf hoher Ebene finden Sie unter. [Hochladen eines Objekts mit Multipart-Upload](#)

Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ermitteln, rufen Sie die Objektmetadaten ab, indem Sie die Methode [Aws\S3\S3Client::headObject\(\)](#) aufrufen, wie im folgenden PHP Codebeispiel gezeigt.

```
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Check which server-side encryption algorithm is used.
```

```
$result = $s3->headObject([
    'Bucket' => $bucket,
    'Key'     => $keyname,
]);
echo $result['ServerSideEncryption'];
```

Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ändern, erstellen Sie mithilfe der Methode [Aws\S3\S3Client::copyObject\(\)](#) eine Kopie des Objekts und löschen Sie das Quellobjekt. Standardmäßig verschlüsselt `copyObject()` das Ziel nicht, es sei denn, Sie fordern explizit die serverseitige Verschlüsselung des Zielobjekts an, indem Sie den Parameter `ServerSideEncryption` mit dem Wert `AES256` angeben. Das folgende PHP Codebeispiel erstellt eine Kopie eines Objekts und fügt dem kopierten Objekt serverseitige Verschlüsselung hinzu.

```
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';

$targetBucket = '*** Your Target Bucket Name ***';
$targetKeyname = '*** Your Target Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Copy an object and add server-side encryption.
$s3->copyObject([
    'Bucket'           => $targetBucket,
    'Key'              => $targetKeyname,
    'CopySource'       => "$sourceBucket/$sourceKeyname",
    'ServerSideEncryption' => 'AES256',
]);
```

Weitere Informationen finden Sie unter den folgenden Themen:

- [AWS SDK for PHP für Amazon S3 Aws\S3\S3Client-Klasse](#)

- [AWS SDK for PHP Dokumentation](#)

Ruby

Bei der Verwendung der AWS SDK for Ruby Um ein Objekt hochzuladen, können Sie mit SSE -S3 angeben, dass das Objekt im Ruhezustand verschlüsselt gespeichert wird. Wenn Sie das Objekt zurücklesen, wird es automatisch entschlüsselt.

Folgendes AWS SDK for Ruby Das Beispiel für Version 3 zeigt, wie angegeben wird, dass eine auf Amazon S3 hochgeladene Datei im Ruhezustand verschlüsselt wird.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)
```

```
puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
#{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Das folgende Beispiel zeigt, wie Sie den Verschlüsselungsstatus eines vorhandenen Objekts bestimmen.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
  # successful; otherwise nil.
  def get_object
    @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
    object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? "no" :
  obj_data.server_side_encryption
end
```

```
puts "Object #{object_key} uses #{encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Wenn die serverseitige Verschlüsselung für das in Amazon S3 gespeicherte Objekt nicht verwendet wird, gibt die Methode `null` zurück.

Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ändern, erstellen Sie eine Kopie des Objekts und löschen dann das Quellobjekt. Standardmäßig verschlüsseln die Methoden zum Kopieren das Ziel nicht, es sei denn, Sie fordern explizit die serverseitige Verschlüsselung an. Sie können die Verschlüsselung des Zielobjekts anfordern, indem Sie den Wert `server_side_encryption` im Hash-Argument der Option angeben, wie im folgenden Ruby-Codebeispiel gezeigt. Das Codebeispiel zeigt, wie ein Objekt kopiert und die Kopie mit SSE-S3 verschlüsselt wird.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
  # used as the source object for
  #
  #           copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the target
  # key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
  # object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
  # nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
```



```
puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
end
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key} and "\
      "encrypted the target with #{target_object.server_side_encryption}
encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Verwendung der AWS CLI

Um SSE -S3 anzugeben, wenn Sie ein Objekt hochladen, indem Sie AWS CLI, verwenden Sie das folgende Beispiel.

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key object-key-name --server-side-encryption AES256 --body file path
```

Weitere Informationen finden Sie unter [put-object](#) im AWS CLI Referenz. Um SSE -S3 anzugeben, wenn Sie ein Objekt kopieren, indem Sie AWS CLI, siehe Objekt [kopieren](#).

Die Verwendung von AWS CloudFormation

Beispiele für die Einrichtung der Verschlüsselung finden Sie mit AWS CloudFormation, siehe [Einen Bucket mit Standardverschlüsselung erstellen](#) und [Einen Bucket erstellen mit AWS](#)

[KMS](#) Beispiel für serverseitige Verschlüsselung mit einem S3-Bucket-Key im AWS::S3::Bucket `ServerSideEncryptionRule` Thema im AWS CloudFormation Benutzerleitfaden.

Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel (SSE-KMS)

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Serverseitige Verschlüsselung ist die Verschlüsselung von Daten am Zielort durch die Anwendung oder den Service, der sie erhält.

Amazon S3 aktiviert automatisch die serverseitige Verschlüsselung mit Amazon S3 S3-verwalteten Schlüsseln (SSE-S3) für das Hochladen neuer Objekte.

Sofern Sie nichts anderes angeben, verwenden Buckets standardmäßig SSE -S3, um Objekte zu verschlüsseln. Sie können sich jedoch dafür entscheiden, Buckets so zu konfigurieren, dass sie serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) stattdessen Schlüssel (SSE-KMS). Weitere Informationen finden Sie unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#).

AWS KMS ist ein Dienst, der sichere, hochverfügbare Hardware und Software kombiniert, um ein für die Cloud skaliertes Schlüsselverwaltungssystem bereitzustellen. Amazon S3 verwendet serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) um Ihre S3-Objektdaten zu verschlüsseln. Wenn SSE - für das Objekt angefordert KMS wird, wird außerdem die S3-Prüfsumme (als Teil der Metadaten des Objekts) in verschlüsselter Form gespeichert. Weitere Informationen zur Prüfsumme finden Sie unter [Überprüfung der Objektintegrität](#).

Wenn Sie KMS Schlüssel verwenden, können Sie AWS KMS durch die [AWS Management Console](#) oder die [AWS KMS API](#) um Folgendes zu tun:

- Schlüssel zentral erstellen, anzeigen, bearbeiten, überwachen, aktivieren oder deaktivieren, rotieren und das Löschen von KMS Schlüsseln planen.
- Definieren Sie die Richtlinien, die steuern, wie und von wem KMS Schlüssel verwendet werden können.
- Prüfen Sie deren Verwendung, um zu beweisen, dass sie korrekt verwendet werden. Das Auditing wird unterstützt von [AWS KMS API](#), aber nicht von [AWS KMSAWS Management Console](#).

Die Sicherheitskontrollen in AWS KMS kann Ihnen helfen, die Compliance-Anforderungen im Zusammenhang mit der Verschlüsselung zu erfüllen. Sie können diese KMS Schlüssel verwenden, um Ihre Daten in Amazon S3 S3-Buckets zu schützen. Wenn Sie SSE - KMS Verschlüsselung mit einem S3-Bucket verwenden, AWS KMS keys muss sich in derselben Region wie der Bucket befinden.

Für die Nutzung fallen zusätzliche Gebühren an AWS KMS keys Weitere Informationen finden Sie unter [.AWS KMS key Konzepte](#) in der AWS Key Management Service Entwicklerhandbuch und [AWS KMS Preisgestaltung](#).

Berechtigungen

Um ein Objekt hochzuladen, das mit einem verschlüsselt ist AWS KMS key für Amazon S3 benötigen Sie `kms:GenerateDataKey` Berechtigungen für den Schlüssel. Um ein Objekt herunterzuladen, das mit einem verschlüsselt ist AWS KMS key, Sie benötigen `kms:Decrypt` Berechtigungen. Für Informationen über AWS KMS Berechtigungen, die für mehrteilige Uploads erforderlich sind, finden Sie unter [Mehrteiliger Upload API und Berechtigungen](#)

Important

Prüfen Sie sorgfältig die Berechtigungen, die Ihnen in Ihren KMS wichtigsten Richtlinien gewährt werden. Beschränken Sie die vom Kunden verwalteten Berechtigungen für KMS wichtige Richtlinien immer nur auf die IAM Hauptbenutzer und AWS Dienste, die auf die entsprechenden zugreifen müssen AWS KMS zentrale Maßnahme. Weitere Informationen finden Sie unter [Wichtige Richtlinien in AWS KMS](#).

Themen

- [AWS KMS keys](#)

- [Amazon-S3-Bucket-Schlüssel](#)
- [Erzwingen serverseitiger Verschlüsselung](#)
- [Verschlüsselungskontext](#)
- [Senden von Anfragen für AWS KMS verschlüsselte Objekte](#)
- [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#)
- [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#)

AWS KMS keys

Wenn Sie serverseitige Verschlüsselung verwenden mit AWS KMS (SSE-KMS), Sie können die Standardeinstellung verwenden [AWS verwalteter Schlüssel](#), oder Sie können einen vom [Kunden verwalteten Schlüssel](#) angeben, den Sie bereits erstellt haben. AWS KMS unterstützt die Verschlüsselung von Umschlägen. S3 verwendet die AWS KMS Funktionen für die Verschlüsselung von Umschlägen, um Ihre Daten weiter zu schützen. Bei der Umschlagverschlüsselung werden Ihre Klartextdaten mit einem Datenschlüssel und anschließend dieser Datenschlüssel mit einem Schlüssel verschlüsselt. KMS Weitere Informationen zur Umschlagverschlüsselung finden Sie unter [Umschlagverschlüsselung](#) im AWS Key Management Service Entwicklerhandbuch.

Wenn Sie keinen vom Kunden verwalteten Schlüssel angeben, erstellt Amazon S3 automatisch einen Von AWS verwalteter Schlüssel in Ihrem AWS-Konto das erste Mal, dass Sie ein mit SSE - verschlüsseltes Objekt KMS zu einem Bucket hinzufügen. Standardmäßig verwendet Amazon S3 diesen KMS Schlüssel für SSE -KMS.

Note

Objekte, die mit SSE - KMS mit verschlüsselt wurden [Von AWS verwaltete Schlüssel](#) kann nicht kontoübergreifend geteilt werden. Wenn Sie KMS Daten kontenübergreifend SSE teilen möchten, müssen Sie einen vom [Kunden verwalteten](#) Schlüssel von verwenden AWS KMS.

Wenn Sie einen vom Kunden verwalteten Schlüssel für SSE - verwenden möchten KMS, erstellen Sie vor der Konfiguration SSE von - einen vom Kunden verwalteten Schlüssel für symmetrische Verschlüsselung. KMS Wenn Sie dann SSE - KMS für Ihren Bucket konfigurieren, geben Sie den vorhandenen, vom Kunden verwalteten Schlüssel an. Weitere Informationen zum symmetrischen Verschlüsselungsschlüssel finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Durch die Erstellung eines vom Kunden verwalteten Schlüssels erhalten Sie mehr Flexibilität und Kontrolle. Beispielsweise können Sie kundenverwaltete Schlüssel erstellen, drehen und deaktivieren. Sie können auch Zugriffskontrollen definieren und die vom Kunden verwalteten Schlüssel prüfen, mit denen Sie Ihre Daten schützen. Weitere Informationen zu den vom Kunden verwalteten und AWS verwaltete Schlüssel finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) in der AWS Key Management Service Leitfaden für Entwickler.

Note

Wenn Sie serverseitige Verschlüsselung mit einem vom Kunden verwalteten Schlüssel verwenden, der in einem externen Schlüsselspeicher gespeichert ist, sind Sie im Gegensatz zu KMS Standardschlüsseln dafür verantwortlich, die Verfügbarkeit und Haltbarkeit Ihres Schlüsselmaterials sicherzustellen. Weitere Informationen zu externen Schlüsselspeichern und wie sie das Modell der gemeinsamen Verantwortung verändern, finden Sie unter [Externe Schlüsselspeicher](#) in der AWS Key Management Service Leitfaden für Entwickler.

Verwendung SSE — KMS Verschlüsselung für kontoübergreifende Operationen

Beachten Sie Folgendes, wenn Sie kontoübergreifende Operationen verschlüsseln:

- Wenn ein AWS KMS key Der Amazon-Ressourcenname (ARN) oder Alias wird bei der Anfrage oder über die Standardverschlüsselungskonfiguration des Buckets, die Von AWS verwalteter Schlüssel (`aws/s3`) wird verwendet.
- Wenn Sie S3-Objekte hochladen oder auf sie zugreifen, indem Sie AWS Identity and Access Management (IAM) Prinzipale, die sich im selben befinden AWS-Konto als KMS Schlüssel können Sie den verwenden Von AWS verwalteter Schlüssel (`aws/s3`).
- Verwenden Sie einen vom Kunden verwalteten Schlüssel, wenn Sie kontoübergreifenden Zugriff auf Ihre S3-Objekte gewähren möchten. Sie können die Richtlinie eines vom Kunden verwalteten Schlüssel so konfigurieren, dass der Zugriff von einem anderen Konto aus möglich ist.
- Wenn Sie einen vom Kunden verwalteten KMS Schlüssel angeben, empfehlen wir die Verwendung eines vollqualifizierten KMS SchlüsselsARN. Wenn Sie stattdessen einen KMS Schlüsselalias verwenden, AWS KMS löst den Schlüssel im Konto des Anfragenden auf. Dieses Verhalten kann dazu führen, dass Daten mit einem KMS Schlüssel verschlüsselt werden, der dem Anforderer und nicht dem Bucket-Besitzer gehört.
- Sie müssen einen Schlüssel angeben, für den Ihnen (dem Anforderer) die Berechtigung Encrypt erteilt wurde. Weitere Informationen finden Sie unter [Erlauben Sie Schlüsselbenutzern, einen](#)

[KMS Schlüssel für kryptografische Operationen zu verwenden](#) in AWS Key Management Service Leitfaden für Entwickler.

Weitere Informationen darüber, wann Sie vom Kunden verwaltete Schlüssel verwenden sollten, und AWS verwaltete KMS Schlüssel finden Sie unter [Sollte ich einen verwenden Von AWS verwalteter Schlüssel oder ein vom Kunden verwalteter Schlüssel zur Verschlüsselung meiner Objekte in Amazon S3?](#)

SSE- KMS Verschlüsselungsworkflow

Wenn Sie sich dafür entscheiden, Ihre Daten mit einem zu verschlüsseln Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel, AWS KMS und Amazon S3 führen die folgenden Envelope-Verschlüsselungsaktionen durch:

1. Amazon S3 fordert einen [Klartext-Datenschlüssel](#) und eine Kopie des Schlüssels an, der unter dem angegebenen KMS Schlüssel verschlüsselt wurde.
2. AWS KMS generiert einen Datenschlüssel, verschlüsselt ihn unter dem KMS Schlüssel und sendet sowohl den Klartext-Datenschlüssel als auch den verschlüsselten Datenschlüssel an Amazon S3.
3. Amazon S3 verschlüsselt die Daten mit dem Datenschlüssel und entfernt anschließend den Klartextschlüssel schnellstmöglich aus dem Arbeitsspeicher.
4. Amazon S3 speichert den verschlüsselten Datenschlüssel im Metadatenformat zusammen mit den verschlüsselten Daten.

Wenn Sie beantragen, dass Ihre Daten entschlüsselt werden, Amazon S3 und AWS KMS führen Sie die folgenden Aktionen aus:

1. Amazon S3 sendet den verschlüsselten Datenschlüssel an AWS KMS in einer Decrypt Anfrage.
2. AWS KMS entschlüsselt den verschlüsselten Datenschlüssel mit demselben KMS Schlüssel und gibt den Klartext-Datenschlüssel an Amazon S3 zurück.
3. Amazon S3 entschlüsselt die verschlüsselten Daten mit dem Klartext-Datenschlüssel und entfernt den Klartext-Datenschlüssel anschließend schnellstmöglich aus dem Arbeitsspeicher.

Important

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3

unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Prüfung SSE — KMS Verschlüsselung

Um Anfragen zu identifizieren, die SSE - angeben KMS, können Sie die Metriken All SSE - KMS requests und % all SSE - KMS requests in Amazon S3 Storage Lens Metriken verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können auch die Buckets SSE - KMS enabled bucket count und % SSE - KMS enabled verwenden, um die Anzahl der Buckets zu ermitteln, die (SSE-KMS) für die [Standard-Bucket-Verschlüsselung](#) verwendet werden. Weitere Informationen finden Sie unter [Bewertung Ihrer Speicheraktivität und -nutzung mit S3 Storage Lens](#). Eine vollständige Liste der Metriken finden Sie im [Glossar der S3-Storage-Lens-Metriken](#).

Um die Nutzung Ihrer zu überprüfen AWS KMS Schlüssel für Ihre SSE - KMS verschlüsselten Daten, die Sie verwenden können AWS CloudTrail Logs. Sie können Einblick in Ihre [kryptografischen Operationen](#) erhalten, wie zum Beispiel [GenerateDataKey](#) und [Decrypt](#). CloudTrail unterstützt zahlreiche [Attributwerte](#) zum Filtern Ihrer Suche, einschließlich Ereignisname, Benutzername und Ereignisquelle.

Amazon-S3-Bucket-Schlüssel

Wenn Sie die serverseitige Verschlüsselung konfigurieren mit AWS KMS (SSE-KMS), Sie können Ihre Buckets so konfigurieren, dass sie S3-Bucket-Keys für SSE - verwenden. KMS Die Verwendung eines Schlüssels auf Bucket-Ebene für SSE - KMS kann Ihre AWS KMS Anforderungskosten um bis zu 99 Prozent durch Verringerung des Anforderungsverkehrs von Amazon S3 auf AWS KMS.

Wenn Sie einen Bucket so konfigurieren, dass er einen S3-Bucket-Key für SSE — KMS für neue Objekte verwendet, AWS KMS generiert einen Schlüssel auf Bucket-Ebene, der verwendet wird, um eindeutige [Datenschlüssel](#) für Objekte im Bucket zu erstellen. Dieser S3-Bucket Key wird für einen begrenzten Zeitraum innerhalb von Amazon S3 verwendet, wodurch Amazon S3 weiter weniger Anfragen stellen muss AWS KMS um Verschlüsselungsvorgänge abzuschließen. Weitere Informationen zur Verwendung von S3-Bucket-Schlüsseln finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Erzwingen serverseitiger Verschlüsselung

Wenn Sie die serverseitige Verschlüsselung aller Objekte in einem bestimmten Amazon-S3-Bucket anfordern möchten, können Sie eine Bucket-Richtlinie verwenden. Die folgende Bucket-Richtlinie verweigert beispielsweise allen Benutzern die Berechtigung Objekt (`s3:PutObject`) hochladen, wenn die Anfrage keinen `x-amz-server-side-encryption-aws-kms-key-id` Header enthält, der eine serverseitige Verschlüsselung mit SSE - anfordert. KMS

```
{
  "Version":"2012-10-17",
  "Id":"PutObjectPolicy",
  "Statement":[{
    "Sid":"DenyObjectsThatAreNotSSEKMS",
    "Effect":"Deny",
    "Principal":"*",
    "Action":"s3:PutObject",
    "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*",
    "Condition":{"
      "Null":{"
        "s3:x-amz-server-side-encryption-aws-kms-key-id":"true"
      }
    }
  }
]
```

Um das zu verlangen, ist ein bestimmter AWS KMS key Um die Objekte in einem Bucket zu verschlüsseln, können Sie den `s3:x-amz-server-side-encryption-aws-kms-key-id` Bedingungsschlüssel verwenden. Um den KMS Schlüssel anzugeben, müssen Sie einen Schlüssel Amazon Resource Name (ARN) verwenden, der das `arn:aws:kms:region:acct-id:key/key-id` Format hat. AWS Identity and Access Management überprüft nicht, ob die Zeichenfolge für `s3:x-amz-server-side-encryption-aws-kms-key-id` existiert.

Note

Wenn Sie ein Objekt hochladen, können Sie den KMS Schlüssel mithilfe des `x-amz-server-side-encryption-aws-kms-key-id` Headers angeben oder sich auf Ihre [Standardkonfiguration für die Bucket-Verschlüsselung](#) verlassen. Wenn Ihre `PutObject` Anfrage `aws:kms` im `x-amz-server-side-encryption` Header spezifiziert ist, aber nicht

den `x-amz-server-side-encryption-aws-kms-key-id` Header spezifiziert, dann geht Amazon S3 davon aus, dass Sie den Von AWS verwalteter Schlüssel. Unabhängig davon, die AWS KMS Die Schlüssel-ID, die Amazon S3 für die Objektverschlüsselung verwendet, muss mit der übereinstimmen AWS KMS Schlüssel-ID in der Richtlinie, andernfalls lehnt Amazon S3 die Anfrage ab.

Eine vollständige Liste der Amazon S3-spezifischen Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Verschlüsselungskontext

Ein Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten können. Der Verschlüsselungskontext ist nicht verschlüsselt. Wenn für eine Verschlüsselungsoperation ein Verschlüsselungskontext angegeben wird, muss Amazon S3 denselben Verschlüsselungskontext auch für die Entschlüsselungsoperation angeben. Andernfalls schlägt die Entschlüsselung fehlschlägt. AWS KMS verwendet den Verschlüsselungskontext als [zusätzliche authentifizierte Daten](#) (AAD), um die [authentifizierte](#) Verschlüsselung zu unterstützen. Weitere Informationen zum Verschlüsselungskontext finden Sie unter [Verschlüsselungskontext im](#) AWS Key Management Service Entwicklerhandbuch.

Standardmäßig verwendet Amazon S3 das Objekt oder den Bucket Amazon Resource Name (ARN) als Verschlüsselungskontextpaar:

- Wenn Sie SSE - verwenden, KMS ohne einen S3-Bucket Key zu aktivieren, ARN wird das Objekt als Verschlüsselungskontext verwendet.

```
arn:aws:s3:::object_ARN
```

- Wenn Sie SSE - verwenden KMS und einen S3-Bucket-Key aktivieren, ARN wird der Bucket als Verschlüsselungskontext verwendet. Weitere Informationen zu S3-Bucket-Schlüsseln finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

```
arn:aws:s3:::bucket_ARN
```

Sie können optional ein zusätzliches Verschlüsselungskontextpaar bereitstellen, indem Sie den `x-amz-server-side-encryption-context` Header in einer [s3: PutObject](#) -Anforderung

verwenden. Da der Verschlüsselungskontext jedoch nicht verschlüsselt ist, sollte er keine sensiblen Informationen enthalten. Amazon S3 speichert dieses zusätzliche Schlüsselpaar zusammen mit dem Standardverschlüsselungskontext. Wenn Ihre PUT-Anforderung verarbeitet wird, hängt Amazon S3 den Standardverschlüsselungskontext `aws:s3:arn` an den von Ihnen bereitgestellten an.

Sie können den Verschlüsselungskontext verwenden, um Ihre kryptografischen Vorgänge zu identifizieren und zu kategorisieren. Sie können auch den ARN Standardwert für den Verschlüsselungskontext verwenden, um relevante Anfragen in zu verfolgen AWS CloudTrail indem Sie sich ansehen, welcher Amazon S3 mit welchem Verschlüsselungsschlüssel verwendet ARN wurde.

Im `requestParameters` Feld einer CloudTrail Protokolldatei sieht der Verschlüsselungskontext dem folgenden ähnlich.

```
"encryptionContext": {
  "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1/file_name"
}
```

Wenn Sie SSE - KMS mit der optionalen Funktion S3 Bucket Keys verwenden, entspricht der Wert des Verschlüsselungskontextes dem Wert ARN des Buckets.

```
"encryptionContext": {
  "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1"
}
```

Senden von Anfragen für AWS KMS verschlüsselte Objekte

Important

Alles GET und PUT Anfragen für AWS KMS verschlüsselte Objekte müssen mit Secure Sockets Layer (SSL) oder Transport Layer Security (TLS) erstellt werden. Anfragen müssen außerdem mit gültigen Anmeldeinformationen signiert werden, z. B. AWS Signaturversion 4 (oder AWS Signaturversion 2).

AWS Signature Version 4 ist der Prozess des Hinzufügens von Authentifizierungsinformationen zu AWS Anfragen gesendet vonHTTP. Aus Sicherheitsgründen richten sich die meisten Anfragen an AWS müssen mit einem Zugriffsschlüssel signiert sein, der aus einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel besteht. Diese beiden Schlüssel werden in der Regel

als Sicherheitsanmeldeinformationen bezeichnet. Weitere Informationen finden Sie unter [Authentifizieren von Anfragen \(AWSSignaturprozess für Signature Version 4 und Signature Version 4\)](#).

⚠ Important

Wenn Ihr Objekt SSE - verwendetKMS, senden Sie keine Header für GET Verschlüsselungsanfragen für Anfragen und HEAD Anfragen. Andernfalls erhalten Sie den Fehler HTTP400 Bad Request.

Themen

- [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#)
- [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#)

Geben Sie die serverseitige Verschlüsselung an mit AWS KMS (SSE-KMS)

⚠ Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder SSE - KMS verwenden. DSSE KMS

Sie können die Verschlüsselung anwenden, wenn Sie entweder ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren.

Sie können SSE KMS mithilfe der Amazon S3 S3-Konsole REST API angeben, AWS SDKs, und die AWS Command Line Interface (AWS CLI). Weitere Informationen finden Sie in den folgenden Themen.

Note

Sie können Multiregion verwenden AWS KMS keys auf Amazon S3. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter [Verwenden von Schlüsseln für mehrere Regionen in AWS Key Management Service Entwicklerhandbuch](#).

Note

Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie die Erlaubnis, den Schlüssel zu verwenden. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [Erstellen von KMS Schlüsseln, die andere Konten verwenden können](#), in AWS Key Management Service Leitfaden für Entwickler.

Verwenden der S3-Konsole

In diesem Thema wird beschrieben, wie Sie den Verschlüsselungstyp eines Objekts für die serverseitige Verschlüsselung festlegen oder ändern AWS Key Management Service (AWS KMS) Tasten (SSE-KMS) mithilfe der Amazon S3 S3-Konsole.

 Note

- Wenn Sie die Verschlüsselung eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).
- Wenn Sie den Verschlüsselungstyp für ein Objekt mit benutzerdefinierten Tags ändern, benötigen Sie die entsprechende `s3:GetObjectTagging` Genehmigung. Wenn Sie den Verschlüsselungstyp für ein Objekt ändern, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Genehmigung.

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, wird der Verschlüsselungstyp für das Objekt aktualisiert, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

So fügen Sie die Verschlüsselung für ein Objekt hinzu oder ändern Sie sie

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
4. Wählen Sie in der Liste Objekte den Namen des Objekts aus, für das Sie eine Verschlüsselung hinzufügen oder ändern möchten.

Die Detailseite des Objekts wird angezeigt. Sie enthält mehrere Abschnitte mit den Eigenschaften des Objekts.

5. Wählen Sie die Registerkarte Eigenschaften aus.
6. Scrollen Sie nach unten zum Abschnitt Serverseitige Verschlüsselungseinstellungen und wählen Sie Bearbeiten aus.

Die Seite Serverseitige Verschlüsselung bearbeiten wird geöffnet.

7. Wählen Sie unter Serverseitige Verschlüsselung für Verschlüsselungseinstellungen die Option Überschreiben der Standardverschlüsselungs-Bucket-Einstellungen aus.

- Wählen Sie unter Verschlüsselungstyp die Option Serverseitige Verschlüsselung mit AWS Key Management Service Schlüssel (SSE-KMS).

 **Important**

Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, gelten für Sie die Quoten für Anfragen pro Sekunde (RPS) von AWS KMS. Für weitere Informationen über AWS KMS Kontingente und wie Sie eine Quotenerhöhung beantragen können, finden Sie unter [Kontingente](#) im AWS Key Management Service Leitfaden für Entwickler.

- Unter AWS KMS Schlüssel, führen Sie einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie dann Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

 **Important**

Sie können nur KMS Schlüssel verwenden, die in derselben Version verfügbar sind AWS-Region wie der Eimer. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört,

benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN.

Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter [Identifizieren symmetrischer und KMS asymmetrischer Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

10. Wählen Sie Änderungen speichern.

Note

Diese Aktion wendet auf alle angegebenen Objekte Verschlüsselung an. Warten Sie beim Verschlüsseln von Ordnern, bis die Speicheroperation abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen.

Mit dem REST API

Wenn Sie ein Objekt erstellen, d. h. wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die Verwendung der serverseitigen Verschlüsselung mit angeben AWS KMS keys (SSE-) um Ihre Daten zu verschlüsseln. Fügen Sie hierzu der Anforderung den Header `x-amz-server-side-encryption` hinzu. Setzen Sie den Wert des Headers auf den `aws:kms`-Verschlüsselungsalgorithmus. Amazon S3 bestätigt mit SSE —, dass Ihr Objekt gespeichert wurde, KMS indem es den Antwort-Header zurückgibt `x-amz-server-side-encryption`.

Wenn Sie den Header `x-amz-server-side-encryption` mit dem Wert `aws:kms` angeben, können Sie auch die folgenden Anforderungs-Header verwenden:

- `x-amz-server-side-encryption-aws-kms-key-id`
- `x-amz-server-side-encryption-context`
- `x-amz-server-side-encryption-bucket-key-enabled`

Themen

- [Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS](#)
- [Verschlüsselungskontext \(x-amz-server-side-encryption-context\)](#)

- [AWS KMS Schlüssel-ID \(x-amz-server-side-encryption-aws-kms-key-id\)](#)
- [S3-Bucket-Schlüssel \(x-amz-server-side-encryption-aws-bucket-key-enabled\)](#)

Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS

Die folgenden REST API Operationen akzeptieren die Header `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, und `x-amz-server-side-encryption-context` Request-Header.

- [PutObject](#)— Wenn Sie Daten mithilfe der PUT API Operation hochladen, können Sie diese Anforderungsheader angeben.
- [CopyObject](#) – Wenn Sie ein Objekt kopieren, erhalten Sie ein Quell- und ein Zielobjekt. Wenn Sie bei der CopyObject Operation SSE KMS -Header übergeben, werden sie nur auf das Zielobjekt angewendet. Wenn Sie ein vorhandenes Objekt kopieren, unabhängig davon, ob das Quellobjekt verschlüsselt ist oder nicht, wird das Zielobjekt nicht verschlüsselt, es sei denn, Sie fordern ausdrücklich serverseitige Verschlüsselung an.
- [POST Object](#)— Wenn Sie einen POST Vorgang verwenden, um ein Objekt hochzuladen, geben Sie anstelle der Anforderungsheader dieselben Informationen in die Formularfelder ein.
- [CreateMultipartUpload](#)— Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie diese Header angeben. Sie geben diese Header in der Anfrage an.
`CreateMultipartUpload`

Die Antwortheader der folgenden REST API Operationen geben den `x-amz-server-side-encryption` Header zurück, wenn ein Objekt mithilfe serverseitiger Verschlüsselung gespeichert wird.

- [PutObject](#)
- [CopyObject](#)
- [POST Object](#)
- [CreateMultipartUpload](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [CompleteMultipartUpload](#)
- [GetObject](#)

- [HeadObject](#)

⚠ Important

- Alle GET und PUT Anfragen für ein Objekt, geschützt durch AWS KMS schlagen fehl, wenn Sie diese Anfragen nicht mithilfe von Secure Sockets Layer (SSL), Transport Layer Security (TLS) oder Signature Version 4 stellen.
- Wenn Ihr Objekt SSE - verwendet KMS, senden Sie keine Header für GET Verschlüsselungsanfragen für Anfragen und HEAD Anfragen, da Sie sonst die BadRequest Fehlermeldung HTTP400 erhalten.

Verschlüsselungskontext (`x-amz-server-side-encryption-context`)

Wenn Sie angeben `x-amz-server-side-encryption:aws:kms`, API unterstützt Amazon S3 einen Verschlüsselungskontext mit dem `x-amz-server-side-encryption-context` Header. Ein Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten können.

Amazon S3 verwendet automatisch das Objekt oder den Bucket Amazon Resource Name (ARN) als Verschlüsselungskontextpaar. Wenn Sie SSE - verwenden, KMS ohne einen S3-Bucket Key zu aktivieren, verwenden Sie das Objekt ARN als Ihren Verschlüsselungskontext; zum Beispiel `arn:aws:s3:::object_ARN`. Wenn Sie jedoch SSE - verwenden KMS und einen S3-Bucket-Key aktivieren, verwenden Sie den Bucket ARN für Ihren Verschlüsselungskontext; zum Beispiel `arn:aws:s3:::bucket_ARN`.

Sie können optional ein zusätzliches Verschlüsselungskontextpaar bereitstellen, indem Sie den Header `x-amz-server-side-encryption-context` verwenden. Da der Verschlüsselungskontext jedoch nicht verschlüsselt ist, stellen Sie sicher, dass er keine vertraulichen Informationen enthält. Amazon S3 speichert dieses zusätzliche Schlüsselpaar zusammen mit dem Standardverschlüsselungskontext.

Weitere Informationen zum Verschlüsselungskontext in Amazon S3 finden Sie unter [Verschlüsselungskontext](#). Allgemeine Informationen zum Verschlüsselungskontext finden Sie unter [AWS Key Management Service Konzepte — Verschlüsselungskontext](#) im AWS Key Management Service Leitfaden für Entwickler.

AWS KMS Schlüssel-ID (**x-amz-server-side-encryption-aws-kms-key-id**)

Sie können den Header `x-amz-server-side-encryption-aws-kms-key-id` verwenden, um die ID des vom Kunden verwalteten Schlüssels anzugeben, der zum Schutz der Daten verwendet wird. Wenn Sie den `x-amz-server-side-encryption:aws:kms` Header angeben, den `x-amz-server-side-encryption-aws-kms-key-id` Header aber nicht angeben, verwendet Amazon S3 den Von AWS verwalteter Schlüssel (`aws/s3`) um die Daten zu schützen. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden möchten, müssen Sie den `x-amz-server-side-encryption-aws-kms-key-id`-Header des vom Kunden verwalteten Schlüssels angeben.

Important

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

S3-Bucket-Schlüssel (**x-amz-server-side-encryption-aws-bucket-key-enabled**)

Sie können den `x-amz-server-side-encryption-aws-bucket-key-enabled` Anforderungsheader verwenden, um einen S3-Bucket-Key auf Objektebene zu aktivieren oder zu deaktivieren. S3-Bucket-Keys reduzieren Ihre AWS KMS Anforderungskosten durch Verringerung des Anforderungsverkehrs von Amazon S3 auf AWS KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Wenn Sie den Header `x-amz-server-side-encryption:aws:kms`, jedoch nicht den Header `x-amz-server-side-encryption-aws-bucket-key-enabled` angeben, werden die Einstellung des S3-Bucket-Schlüssels für den Ziel-Bucket verwendet, um Ihr Objekt zu verschlüsseln. Weitere Informationen finden Sie unter [Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene](#).

Verwendung der AWS CLI

Um das folgende Beispiel zu verwenden AWS CLI Befehle, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die Verwendung der serverseitigen Verschlüsselung mit angeben AWS KMS Schlüssel zum

Verschlüsseln Ihrer Daten. Fügen Sie hierzu der Anforderung den Header `--server-side-encryption aws:kms` hinzu. Verwenden Sie die `--ssekms-key-id` `example-key-id`, um Ihren [Kunden hinzuzufügen, verwaltet AWS KMS Schlüssel](#), den Sie erstellt haben. Wenn Sie einen angeben `--server-side-encryption aws:kms`, aber nicht angeben AWS KMS Schlüssel-ID, Amazon S3 verwendet eine AWS verwalteter Schlüssel.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-side-encryption aws:kms --ssekms-key-id example-key-id --ssekms-encryption-context example-encryption-context --body filepath
```

Sie können S3 Bucket Keys für Ihre `copy-object` Operationen aktivieren `put-object` oder deaktivieren, indem Sie `--bucket-key-enabled` oder hinzufügen `--no-bucket-key-enabled`. Mit S3 Bucket Keys können Sie Ihre AWS KMS Anforderungskosten durch Verringerung des Anforderungsverkehrs von Amazon S3 auf AWS KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit S3 Bucket Keys](#).

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-side-encryption aws:kms --bucket-key-enabled --body filepath
```

Sie können ein Objekt aus einem Quell-Bucket in einen neuen Bucket kopieren und SSE — KMS Verschlüsselung angeben.

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket/example-object-key --bucket amzn-s3-demo-bucket2 --key example-object-key --server-side-encryption aws:kms --sse-kms-key-id example-key-id --ssekms-encryption-context example-encryption-context
```

Verwendung der AWS SDKs

Bei der Verwendung AWS SDKs, Sie können Amazon S3 zur Nutzung anfordern AWS KMS keys für serverseitige Verschlüsselung. Die folgenden Beispiele zeigen, wie man SSE - KMS mit dem AWS SDKs für Java und .NET. Informationen zu anderen SDKs finden Sie unter [Beispielcode und Bibliotheken](#) auf der AWS Entwicklerzentrum.

Important

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu

diesen Schlüsseln finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

CopyObject-Operation

Beim Kopieren von Objekten fügen Sie dieselben Anforderungseigenschaften (`ServerSideEncryptionMethod` und `ServerSideEncryptionKeyManagementServiceKeyId`) hinzu, um Amazon S3 zur Verwendung eines AWS KMS key. Weitere Hinweise zum Kopieren von Objekten finden Sie unter [Objekte kopieren, verschieben und umbenennen](#).

PUT-Operation

Java

Beim Hochladen eines Objekts mithilfe der AWS SDK for Java, können Sie Amazon S3 auffordern, ein AWS KMS key indem Sie die `SSEAwsKeyManagementParams` Eigenschaft hinzufügen, wie in der folgenden Anfrage gezeigt:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
    keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

In diesem Fall verwendet Amazon S3 den Von AWS verwalteter Schlüssel (`aws/s3`). Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#). Sie können optional einen symmetrischen KMS Verschlüsselungsschlüssel erstellen und diesen in der Anforderung angeben, wie im folgenden Beispiel gezeigt:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
    keyName, file).withSSEAwsKeyManagementParams(new
    SSEAwsKeyManagementParams(keyID));
```

Weitere Informationen zum Erstellen von vom Kunden verwalteten Schlüsseln finden Sie unter [Programmieren der AWS KMS API](#) in der AWS Key Management Service Leitfaden für Entwickler.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter [Objekte hochladen](#).

- Informationen zu mehrteiligen Uploads, bei denen mehrteilige Uploads auf hoher oder niedriger Ebene verwendet werden, finden Sie API unter [Hochladen eines Objekts mit Multipart-Upload](#)

.NET

Beim Hochladen eines Objekts mithilfe der AWS SDK for .NET, können Sie Amazon S3 auffordern, ein AWS KMS key indem Sie die `ServerSideEncryptionMethod` Eigenschaft hinzufügen, wie in der folgenden Anfrage gezeigt:

```
PutObjectRequest putRequest = new PutObjectRequest
{
    BucketName = amzn-s3-demo-bucket,
    Key = keyName,
    // other properties
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS
};
```

In diesem Fall verwendet Amazon S3 den Von AWS verwalteter Schlüssel. Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#). Sie können optional Ihren eigenen, vom Kunden verwalteten Schlüssel für symmetrische Verschlüsselung erstellen und diesen in der Anfrage angeben, wie im folgenden Beispiel gezeigt:

```
PutObjectRequest putRequest1 = new PutObjectRequest
{
    BucketName = amzn-s3-demo-bucket,
    Key = keyName,
    // other properties
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
    ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Weitere Informationen zur Erstellung von vom Kunden verwalteten Schlüsseln finden Sie unter [Programmieren der AWS KMS API](#) in der AWS Key Management Service Leitfaden für Entwickler.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter [Objekte hochladen](#).

- Informationen zu mehrteiligen Uploads, bei denen mehrteilige Uploads auf hoher oder niedriger Ebene verwendet werden, finden Sie API unter [Hochladen eines Objekts mit Multipart-Upload](#)

Vorsigniert URLs

Java

Beim Erstellen eines Vorsignierten URL für ein Objekt, das mit einem verschlüsselt ist AWS KMS key, müssen Sie Signature Version 4 explizit angeben, wie im folgenden Beispiel gezeigt:

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
    new ProfileCredentialsProvider(), clientConfiguration);
...
```

Ein Codebeispiel finden Sie unter [Objekte mit vorsignierten teilen URLs](#).

.NET

Wenn Sie ein URL für ein Objekt vorsigniertes Objekt erstellen, das mit einem verschlüsselt ist AWS KMS key, müssen Sie Signature Version 4 explizit angeben, wie im folgenden Beispiel gezeigt:

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Ein Codebeispiel finden Sie unter [Objekte mit vorsignierten teilen URLs](#).

Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys

Amazon S3 Bucket Keys reduzieren die Kosten für die serverseitige Amazon S3 S3-Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS). Die Verwendung eines Schlüssels auf Bucket-Ebene für SSE - KMS kann reduzieren AWS KMS Anforderungskosten um bis zu 99 Prozent durch Verringerung des Anforderungsverkehrs von Amazon S3 auf AWS KMS. Mit ein paar Klicks in der AWS Management Console, und ohne Änderungen an Ihren Client-Anwendungen vornehmen zu müssen, können Sie Ihren Bucket so konfigurieren, dass er einen S3-Bucket-Schlüssel SSE für die KMS Verschlüsselung neuer Objekte verwendet.

Note

S3-Bucket-Keys werden für die zweischichtige serverseitige Verschlüsselung mit nicht unterstützten AWS Key Management Service (AWS KMS) Schlüsseln (DSSE-KMS).

S3-Bucket-Tasten für SSE - KMS

Workloads, die auf Millionen oder Milliarden von Objekten zugreifen, die mit SSE — verschlüsselt sind, können große Mengen an Anfragen generieren AWS KMS. Wenn Sie SSE — KMS zum Schutz Ihrer Daten ohne einen S3 Bucket Key — verwenden, verwendet Amazon S3 eine individuelle AWS KMS [Datenschlüssel](#) für jedes Objekt. In diesem Fall ruft Amazon S3 an AWS KMS jedes Mal, wenn eine Anfrage gegen ein KMS -verschlüsseltes Objekt gestellt wird. Hinweise zur KMS Funktionsweise von SSE - finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

Wenn Sie Ihren Bucket so konfigurieren, dass er einen S3-Bucket-Key für SSE - verwendet KMS, generiert AWS einen kurzlebigen Schlüssel auf Bucket-Ebene aus AWS KMS und behält ihn dann vorübergehend in S3. Dieser Schlüssel auf Bucket-Ebene erstellt während seines Lebenszyklus einen Datenschlüssel für neue Objekte. S3-Bucket Keys werden für einen begrenzten Zeitraum in Amazon S3 verwendet, sodass S3 weniger Anfragen stellen muss AWS KMS um Verschlüsselungsvorgänge abzuschließen. Dadurch wird der Verkehr von S3 auf reduziert AWS KMS, sodass Sie darauf zugreifen können AWS KMS-verschlüsselte Objekte in Amazon S3 zu einem Bruchteil der bisherigen Kosten.

Eindeutige Schlüssel auf Bucket-Ebene werden mindestens einmal pro Anforderer abgerufen, um sicherzustellen, dass der Zugriff des Anforderers auf den Schlüssel in einem AWS KMS CloudTrail Ereignis. Amazon S3 behandelt Anrufer als unterschiedliche Anforderer, wenn sie unterschiedliche Rollen oder Konten oder dieselbe Rolle mit unterschiedlichen Bereichsrichtlinien verwenden. AWS KMS Die Einsparungen bei Anfragen spiegeln die Anzahl der Anfragenden, das Anforderungsmuster und das relative Alter der angeforderten Objekte wider. Beispielsweise führt eine geringere Anzahl von Anforderern, die mehrere mit demselben Schlüssel auf Bucket-Ebene verschlüsselte Objekte in einem begrenzten Zeitfenster anfordern, zu größeren Einsparungen.

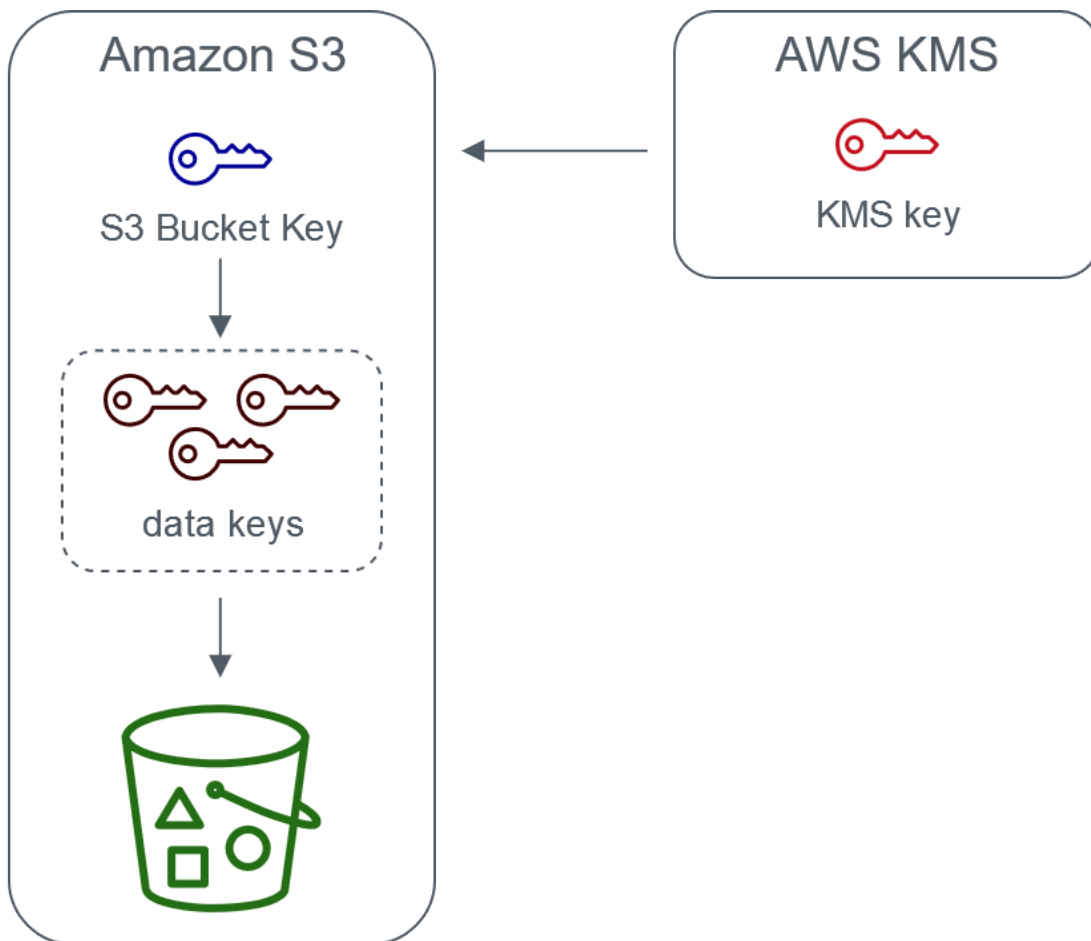
Note

Die Verwendung von S3 Bucket Keys ermöglicht Ihnen Folgendes zu sparen AWS KMS Anforderungskosten durch Reduzierung Ihrer Anfragen an AWS KMS für

EncryptGenerateDataKey, und Decrypt Operationen mithilfe eines Schlüssels auf Bucket-Ebene. Standardmäßig führen nachfolgende Anfragen, die diesen Schlüssel auf Bucket-Ebene nutzen, nicht zu AWS KMS APIAnfragen oder validieren den Zugriff anhand der AWS KMS wichtige Richtlinie.

Wenn Sie einen S3-Bucket-Schlüssel konfigurieren, verwenden Objekte, die sich bereits im Bucket befinden, nicht den S3-Bucket-Schlüssel. Zum Konfigurieren eines S3-Bucket-Schlüssels für vorhandene Objekte können Sie eine CopyObject-Operation verwenden. Weitere Informationen finden Sie unter [Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene](#).

Amazon S3 teilt nur einen S3-Bucket Key für Objekte, die mit demselben verschlüsselt wurden AWS KMS key. S3 Bucket Keys sind kompatibel mit KMS Schlüsseln, die von erstellt wurden AWS KMS, [importiertes Schlüsselmaterial](#) und [Schlüsselmaterial, das durch benutzerdefinierte Schlüsselspeicher unterstützt wird](#).



Server-side encryption with AWS Key Management service using an S3 Bucket Key

Konfigurieren von S3-Bucket-Schlüsseln

Sie können Ihren Bucket so konfigurieren, dass er einen S3-Bucket Key für SSE — KMS für neue Objekte über die Amazon S3 S3-Konsole verwendet. AWS SDKs, AWS CLI, oder RESTAPI. Wenn S3-Bucket Keys in Ihrem Bucket aktiviert sind, verwenden Objekte, die mit einem anderen angegebenen SSE KMS Schlüssel hochgeladen wurden, ihre eigenen S3-Bucket-Keys. Unabhängig von Ihrer S3-Bucket-Schlüsseleinstellung können Sie den Header `x-amz-server-side-encryption-bucket-key-enabled` mit einem `true`- oder `false`-Wert in Ihre Anforderung aufnehmen, um die Bucket-Einstellung zu überschreiben.

Bevor Sie Ihren Bucket für die Verwendung eines S3-Bucket-Schlüssels konfigurieren, lesen Sie [Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten](#).

Konfigurieren eines S3-Bucket-Schlüssels mit der Amazon-S3-Konsole

Wenn Sie einen neuen Bucket erstellen, können Sie Ihren Bucket so konfigurieren, dass er einen S3-Bucket-Key für SSE — KMS für neue Objekte verwendet. Sie können einen vorhandenen Bucket auch so konfigurieren, dass er einen S3-Bucket-Key für SSE — KMS für neue Objekte verwendet, indem Sie Ihre Bucket-Eigenschaften aktualisieren.

Weitere Informationen finden Sie unter [Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet](#).

REST API, AWS CLI, und AWS SDKUnterstützung für S3-Bucket Keys

Sie können das verwenden RESTAPI, AWS CLI, oder AWS SDKum Ihren Bucket so zu konfigurieren, dass er einen S3-Bucket-Key für SSE - KMS für neue Objekte verwendet. Sie können einen S3-Bucket-Schlüssel auch auf Objektebene aktivieren.

Weitere Informationen finden Sie hier:

- [Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene](#)
- [Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet](#)

Die folgenden API Operationen unterstützen S3-Bucket-Keys für SSE -KMS:

- [PutBucketEncryption](#)
 - `ServerSideEncryptionRule` akzeptiert den Parameter `BucketKeyEnabled` zum Aktivieren und Deaktivieren eines S3-Bucket-Schlüssels.

- [GetBucketEncryption](#)
 - `ServerSideEncryptionRule` gibt die Einstellungen für `BucketKeyEnabled` zurück.
- [PutObject](#), [CopyObjectCreateMultipartUpload](#), und [POSTObjekt](#)
 - Der Anforderungs-Header `x-amz-server-side-encryption-bucket-key-enabled` aktiviert oder deaktiviert einen S3-Bucket-Schlüssel auf Objektebene.
- [HeadObject](#), [GetObjectUploadPartCopy](#), [UploadPart](#), und [CompleteMultipartUpload](#)
 - Der Antwort-Header `x-amz-server-side-encryption-bucket-key-enabled` zeigt an, ob ein S3-Bucket-Schlüssel für ein Objekt aktiviert oder deaktiviert ist.

Arbeitet mit AWS CloudFormation

In AWS CloudFormation, die `AWS::S3::Bucket` Ressource enthält eine Verschlüsselungseigenschaft namens `BucketKeyEnabled`, mit der Sie einen S3-Bucket-Key aktivieren oder deaktivieren können.

Weitere Informationen finden Sie unter [Die Verwendung von AWS CloudFormation](#).

Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten

Bevor Sie einen S3-Bucket-Schlüssel aktivieren, beachten Sie bitte die folgenden damit verbundenen Änderungen:

IAModer AWS KMS wichtige Richtlinien

Wenn du existierst AWS Identity and Access Management (IAM) Richtlinien oder AWS KMS Schlüsselrichtlinien verwenden Ihr Objekt Amazon Resource Name (ARN) als Verschlüsselungskontext, um den Zugriff auf Ihren KMS Schlüssel zu verfeinern oder einzuschränken. Diese Richtlinien funktionieren nicht mit einem S3-Bucket Key. S3-Bucket-Keys verwenden den Bucket ARN als Verschlüsselungskontext. Bevor Sie einen S3-Bucket Key aktivieren, aktualisieren Sie Ihre IAM Richtlinien oder AWS KMS wichtige Richtlinien, um Ihren Bucket ARN als Verschlüsselungskontext zu verwenden.

Weitere Informationen zum Verschlüsselungskontext und zu S3-Bucket-Schlüsseln finden Sie unter [Verschlüsselungskontext](#).

CloudTrail Ereignisse für AWS KMS

Nachdem Sie einen S3-Bucket Key aktiviert haben, wird Ihr AWS KMS CloudTrail Ereignisse protokollieren Ihren Bucket ARN anstelle Ihres ObjektsARN. Darüber hinaus werden in Ihren

Protokollen weniger KMS CloudTrail Ereignisse für SSE - KMS Objekte angezeigt. Da wichtige Materialien in Amazon S3 zeitlich begrenzt sind, werden weniger Anfragen gestellt an AWS KMS.

Verwenden eines S3-Bucket-Schlüssels mit Replikation

Sie können S3-Bucket-Keys mit Replikation in derselben Region (SRR) und regionsübergreifender Replikation (CRR) verwenden.

Wenn Amazon S3 ein verschlüsseltes Objekt repliziert, werden im Allgemeinen die Verschlüsselungseinstellungen des Replikatobjekts im Ziel-Bucket beibehalten. Wenn das Quellobjekt jedoch nicht verschlüsselt ist und Ihr Ziel-Bucket eine Standard-Verschlüsselung oder einen S3-Bucket-Schlüssel verwendet, verschlüsselt Amazon S3 das Objekt mit der Konfiguration des Ziel-Buckets.

Die folgenden Beispiele veranschaulichen, wie ein S3-Bucket-Schlüssel mit der Replikation funktioniert. Weitere Informationen finden Sie unter [Replikation verschlüsselter Objekte \(SSE-C, SSE-S3, SSE-KMS\) DSSE KMS](#).

Example Beispiel 1 – Quellobjekt verwendet S3-Bucket-Schlüssel; Ziel-Bucket verwendet Standardverschlüsselung

Wenn Ihr Quellobjekt einen S3-Bucket Key verwendet, Ihr Ziel-Bucket jedoch die Standardverschlüsselung mit SSE - verwendetKMS, behält das Replikatobjekt seine S3-Bucket-Key-Verschlüsselungseinstellungen im Ziel-Bucket bei. Der Ziel-Bucket verwendet weiterhin die Standardverschlüsselung mit SSE -KMS.

Example Beispiel 2 — Das Quellobjekt ist nicht verschlüsselt; der Ziel-Bucket verwendet einen S3-Bucket-Schlüssel mit SSE - KMS

Wenn Ihr Quellobjekt nicht verschlüsselt ist und der Ziel-Bucket einen S3-Bucket-Schlüssel mit SSE - verwendetKMS, wird das Replikat-Objekt mithilfe eines S3-Bucket-Schlüssels mit SSE — KMS im Ziel-Bucket verschlüsselt. Daher unterscheidet sich das ETag des Quell-Objekts von dem ETag des Replikatobjekts. Sie müssen die Anwendungen, die das ETag verwenden, zur Anpassung an diesen Unterschied aktualisieren.

Arbeiten mit S3-Bucket-Schlüsseln

Weitere Informationen zum Aktivieren und Arbeiten mit S3-Bucket-Schlüsseln finden Sie in den folgenden Abschnitten:

- [Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet](#)
- [Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene](#)
- [Anzeigen der Einstellungen für einen S3-Bucket-Schlüssel](#)

Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet

Wenn Sie serverseitige Verschlüsselung konfigurieren mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), können Sie Ihren Bucket so konfigurieren, dass er für neue Objekte einen S3-Bucket-Key für SSE - KMS verwendet. S3-Bucket Keys verringern den Anforderungsverkehr von Amazon S3 auf AWS KMS und reduzieren Sie die Kosten von SSE -KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Sie können Ihren Bucket so konfigurieren, dass er einen S3-Bucket-Key für SSE — KMS für neue Objekte verwendet, indem Sie die Amazon S3 S3-Konsole verwenden RESTAPI, AWS SDKs, AWS Command Line Interface (AWS CLI), oder AWS CloudFormation. Wenn Sie einen S3-Bucket-Key für bestehende Objekte aktivieren oder deaktivieren möchten, können Sie eine CopyObject Operation verwenden. Weitere Informationen finden Sie unter [Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene](#) und [Verwenden von S3-Batch-Vorgänge zum Verschlüsseln von Objekten mit S3-Bucket-Schlüssel](#).

Wenn ein S3-Bucket Key für den Quell- oder Ziel-Bucket aktiviert ist, ist der Verschlüsselungskontext beispielsweise der Bucket Amazon Resource Name (ARN) und nicht das Objekt `ARN:arn:aws:s3:::bucket_ARN`. Sie müssen Ihre IAM Richtlinien aktualisieren, um den Bucket ARN für den Verschlüsselungskontext verwenden zu können. Weitere Informationen finden Sie unter [S3 Bucket-Schlüssel und Replikation](#).

Die folgenden Beispiele veranschaulichen, wie ein S3-Bucket-Schlüssel mit der Replikation funktioniert. Weitere Informationen finden Sie unter [Replikation verschlüsselter Objekte \(SSE-C, SSE -S3, -, SSE -KMS\) DSSE KMS](#).

Voraussetzungen

Bevor Sie Ihren Bucket für die Verwendung eines S3-Bucket-Schlüssels konfigurieren, lesen Sie [Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten](#).

Verwenden der S3-Konsole

In der S3-Konsole können Sie einen S3-Bucket-Schlüssel für einen neuen oder vorhandenen Bucket aktivieren oder deaktivieren. Objekte in der S3-Konsole übernehmen ihre S3-Bucket-Schlüssel-Einstellung aus der Bucket-Konfiguration. Wenn Sie einen S3-Bucket Key für Ihren Bucket aktivieren, verwenden neue Objekte, die Sie in den Bucket hochladen, einen S3-Bucket Key für SSE -KMS.

Hochladen, Kopieren oder Ändern von Objekten in Buckets, für die ein S3-Bucket-Schlüssel aktiviert ist

Wenn Sie ein Objekt in einen Bucket hochladen oder kopieren, für den ein S3-Bucket-Schlüssel aktiviert ist, oder ein Objekt darin ändern, werden die Einstellung des S3-Bucket-Schlüssels für dieses Objekt möglicherweise aktualisiert, um sie an die Bucket-Konfiguration anzupassen.

Wenn für ein Objekt bereits ein S3-Bucket-Schlüssel aktiviert ist, ändern sich die S3-Bucket-Schlüssel-Einstellungen für dieses Objekt nicht, wenn Sie das Objekt kopieren oder ändern. Wenn Sie jedoch ein Objekt ändern oder kopieren, für das kein S3-Bucket-Schlüssel aktiviert ist, und der Ziel-Bucket eine S3-Bucket-Schlüssel-Konfiguration hat, übernimmt das Objekt die S3-Bucket-Schlüssel-Einstellungen des Ziel-Buckets. Wenn beispielsweise für den Ziel-Bucket ein S3-Bucket-Schlüssel aktiviert ist, für das Quellobjekt jedoch nicht, wird ein S3-Bucket-Schlüssel für das Objekt aktiviert.

So aktivieren Sie einen S3-Bucket-Schlüssel beim Erstellen eines neuen Buckets

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.
4. Geben Sie Ihren Bucket-Namen ein und wählen Sie Ihren AWS-Region.
5. Wählen Sie unter Standardverschlüsselung für Verschlüsselungsschlüsseltyp die Option AWS Key Management Service Schlüssel (SSE-KMS).
6. Unter AWS KMS Taste, führen Sie einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:
 - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

7. Wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).
8. Wählen Sie Create Bucket (Bucket erstellen) aus.

Amazon S3 erstellt Ihren Bucket mit einem aktivierten S3-Bucket-Schlüssel. Neue Objekte, die Sie in den Bucket hochladen, verwenden einen S3-Bucket-Schlüssel.

Um einen S3-Bucket-Schlüssel zu deaktivieren, führen Sie die vorherigen Schritte aus und wählen Deaktivieren.

So aktivieren Sie einen S3-Bucket-Schlüssel für einen vorhandenen Bucket

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
6. Wählen Sie unter Standardverschlüsselung für Verschlüsselungsschlüsseltyp AWS Key Management Service Schlüssel (SSE-KMS).
7. Unter AWS KMS Taste, führen Sie einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie dann Ihren KMS-Schlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

8. Wählen Sie unter Bucket-Schlüssel die Option Aktivieren.
9. Wählen Sie Save Changes (Änderungen speichern).

Amazon S3 aktiviert einen S3-Bucket-Schlüssel für neue Objekte, die zu Ihrem Bucket hinzugefügt werden. Bestehende Objekte verwenden den S3-Bucket-Schlüssel nicht. Zum Konfigurieren eines S3-Bucket-Schlüssels für vorhandene Objekte können Sie eine CopyObject-Operation verwenden. Weitere Informationen finden Sie unter [Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene](#) .

Um einen S3-Bucket-Schlüssel zu deaktivieren, führen Sie die vorherigen Schritte aus und wählen Deaktivieren.

Mit dem REST API

Sie können [PutBucketEncryption](#) einen S3-Bucket Key für Ihren Bucket aktivieren oder deaktivieren. Um einen S3-Bucket Key mit zu konfigurierenPutBucketEncryption, verwenden Sie den [ServerSideEncryptionRule](#) Datentyp, der die Standardverschlüsselung mit SSE - beinhaltetKMS. Sie können optional auch einen vom Kunden verwalteten Schlüssel verwenden, indem Sie die KMS Schlüssel-ID für den vom Kunden verwalteten Schlüssel angeben.

Weitere Informationen und eine Beispielsyntax finden Sie unter [PutBucketEncryption](#).

Verwendung der AWS SDK für Java

Das folgende Beispiel aktiviert die Standard-Bucket-Verschlüsselung mit SSE - KMS und einem S3-Bucket-Key mithilfe von AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

ServerSideEncryptionByDefault serverSideEncryptionByDefault = new
    ServerSideEncryptionByDefault()
    .withSSEAlgorithm(SSEAlgorithm.KMS);
ServerSideEncryptionRule rule = new ServerSideEncryptionRule()
    .withApplyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
    .withBucketKeyEnabled(true);
ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
    new ServerSideEncryptionConfiguration().withRules(Collections.singleton(rule));

SetBucketEncryptionRequest setBucketEncryptionRequest = new
    SetBucketEncryptionRequest()
    .withServerSideEncryptionConfiguration(serverSideEncryptionConfiguration)
    .withBucketName(bucketName);

s3client.setBucketEncryption(setBucketEncryptionRequest);
```

Verwendung der AWS CLI

Das folgende Beispiel aktiviert die Standard-Bucket-Verschlüsselung mit SSE - KMS und einem S3-Bucket-Key unter Verwendung von AWS CLI. Ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "aws:kms",
                "KMSMasterKeyID": "KMS-Key-ARN"
            },
            "BucketKeyEnabled": true
        }
    ]
}
```



```
    }  
  ]  
}'
```

Die Verwendung von AWS CloudFormation

Weitere Informationen zur Konfiguration eines S3-Bucket Keys mit AWS CloudFormation, siehe [AWS: :S3: :Bucket ServerSideEncryptionRule](#) im AWS CloudFormation Benutzerleitfaden.

Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene

Wenn Sie eine PUT COPY Oder-Operation mit dem ausführen RESTAPI, AWS SDKs, oder AWS CLI, Sie können einen S3-Bucket-Key auf Objektebene aktivieren oder deaktivieren, indem Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader mit dem `false` Wert `true` oder hinzufügen. S3-Bucket-Keys reduzieren die Kosten der serverseitigen Verschlüsselung mithilfe von AWS Key Management Service (AWS KMS) (SSE-KMS) durch Verringerung des Anforderungsverkehrs von Amazon S3 zu AWS KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Wenn Sie mithilfe einer COPY Operation PUT oder einen S3-Bucket Key für ein Objekt konfigurieren, aktualisiert Amazon S3 nur die Einstellungen für dieses Objekt. Die S3-Bucket-Schlüssel-Einstellungen für den Ziel-Bucket ändern sich nicht. Wenn Sie eine COPY Oder-Anfrage für ein PUT KMS -verschlüsseltes Objekt an einen Bucket mit aktivierten S3-Bucket-Keys senden, verwendet Ihr Vorgang auf Objektebene automatisch S3-Bucket-Keys, sofern Sie die Schlüssel im Anforderungsheader nicht deaktivieren. Wenn Sie keinen S3-Bucket-Schlüssel für Ihr Objekt angeben, wendet Amazon S3 die S3-Bucket-Schlüssel-Einstellungen für den Ziel-Bucket auf das Objekt an.

Voraussetzung:

Bevor Sie Ihr Objekt für die Verwendung eines S3-Bucket-Schlüssels konfigurieren, lesen Sie [Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten](#).

Themen

- [Amazon S3 Batchvorgänge](#)
- [Verwenden von REST API](#)
- [Verwendung der AWS SDK für Java \(PutObject\)](#)
- [Verwendung der AWS CLI \(PutObject\)](#)

Amazon S3 Batchvorgänge

Um Ihre vorhandenen Amazon-S3-Objekte zu verschlüsseln, können Sie Amazon-S3-Batchvorgänge verwenden. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, mit denen gearbeitet werden soll, und Batch Operations ruft die entsprechenden Objekte auf API, um die angegebene Operation auszuführen.

Mit dem [S3-Batch-Vorgangskopiervorgang](#) können Sie vorhandene nicht verschlüsselte Objekte kopieren und sie in denselben Bucket zurückschreiben, wie verschlüsselte Objekte. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen. Weitere Informationen finden Sie unter [Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch](#) und [Encrypting Objects with Amazon S3 Batch Operations \(Verschlüsseln von Objekten mit Amazon S3 Batch Operations\)](#).

Verwenden von REST API

Wenn Sie SSE - verwenden KMS, können Sie einen S3-Bucket-Key für ein Objekt aktivieren, indem Sie die folgenden API Operationen verwenden:

- [PutObject](#)— Wenn Sie ein Objekt hochladen, können Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader angeben, um einen S3-Bucket-Key auf Objektebene zu aktivieren oder zu deaktivieren.
- [CopyObject](#)— Wenn Sie ein Objekt kopieren und SSE - konfigurieren KMS, können Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader angeben, um einen S3-Bucket-Key für Ihr Objekt zu aktivieren oder zu deaktivieren.
- [POSTObjekt](#) — Wenn Sie einen POST Vorgang verwenden, um ein Objekt hochzuladen und SSE - zu konfigurieren KMS, können Sie das `x-amz-server-side-encryption-bucket-key-enabled` Formularfeld verwenden, um einen S3-Bucket-Key für Ihr Objekt zu aktivieren oder zu deaktivieren.
- [CreateMultipartUpload](#)— Wenn Sie mithilfe der CreateMultipartUpload API Operation und der Konfiguration von SSE - große Objekte hochladen KMS, können Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader verwenden, um einen S3-Bucket-Key für Ihr Objekt zu aktivieren oder zu deaktivieren.

Um einen S3-Bucket-Schlüssel auf Objektebene zu aktivieren, schließen Sie den Anforderungs-Header `x-amz-server-side-encryption-bucket-key-enabled` ein. Weitere Informationen zu SSE — KMS und dem REST API finden Sie unter [Mit dem REST API](#).

Verwendung der AWS SDK für Java (PutObject)

Sie können das folgende Beispiel verwenden, um einen S3-Bucket-Key auf Objektebene zu konfigurieren, indem Sie den AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

String bucketName = "amzn-s3-demo-bucket1";
String keyName = "key name for object";
String contents = "file contents";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName,
    contents)
    .withBucketKeyEnabled(true);

s3client.putObject(putObjectRequest);
```

Verwendung der AWS CLI (PutObject)

Sie können Folgendes verwenden AWS CLI Beispiel für die Konfiguration eines S3-Bucket-Keys auf Objektebene als Teil einer PutObject Anfrage.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object key name --server-side-encryption aws:kms --bucket-key-enabled --body filepath
```

Anzeigen der Einstellungen für einen S3-Bucket-Schlüssel

Sie können die Einstellungen für einen S3-Bucket Key auf Bucket- oder Objektebene mithilfe der Amazon S3 S3-Konsole anzeigen RESTAPI, AWS Command Line Interface (AWS CLI), oder AWS SDKs.

S3 Bucket Keys verringern den Anforderungsverkehr von Amazon S3 zu AWS KMS und reduzieren Sie die Kosten für serverseitige Verschlüsselung mit AWS Key Management Service (SSE-KMS). Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Zum Anzeigen der Einstellungen des S3-Bucket-Schlüssels für einen Bucket oder ein Objekt, der/ das die Einstellungen des S3-Bucket-Schlüssels von der Bucket-Konfiguration übernommen hat, benötigen Sie die Berechtigung zum Ausführen der Aktion `s3:GetEncryptionConfiguration`. Weitere Informationen finden Sie [GetBucketEncryption](#) in der Amazon Simple Storage Service API Reference.

Verwenden der S3-Konsole

In der S3-Konsole können Sie die S3-Bucket-Schlüssel-Einstellungen für Ihren Bucket oder Ihr Objekt anzeigen. S3-Bucket-Schlüssel-Einstellungen werden aus der Bucket-Konfiguration übernommen, es sei denn, für die Quellobjekte ist bereits ein S3-Bucket-Schlüssel konfiguriert.

Objekte und Ordner im selben Bucket können unterschiedliche S3-Bucket-Schlüssel-Einstellungen haben. Wenn Sie beispielsweise ein Objekt mit dem hochladen REST API und einen S3-Bucket-Key für das Objekt aktivieren, behält das Objekt seine S3-Bucket-Key-Einstellung im Ziel-Bucket bei, auch wenn S3 Bucket Key im Ziel-Bucket deaktiviert ist. Ein weiteres Beispiel: Wenn Sie einen S3-Bucket-Schlüssel für einen vorhandenen Bucket aktivieren, verwenden Objekte, die sich bereits im Bucket befinden, keinen S3-Bucket-Schlüssel. Für neue Objekte ist jedoch ein S3-Bucket-Schlüssel aktiviert.

So zeigen Sie die Einstellungen des S3-Bucket-Schlüssels für Ihren Bucket an

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.
4. Wählen Sie Properties (Eigenschaften).
5. Im Bereich Standard-Verschlüsselung sehen Sie unter Bucket-Schlüssel die S3-Bucket-Schlüssel-Einstellung für Ihren Bucket.

Wenn Sie die S3-Bucket-Schlüssel-Einstellung nicht sehen, sind Sie möglicherweise nicht berechtigt, die `s3:GetEncryptionConfiguration`-Aktion durchzuführen. Weitere Informationen finden Sie [GetBucketEncryption](#) in der Amazon Simple Storage Service API Reference.

So zeigen Sie die S3-Bucket-Schlüssel-Einstellung für Ihr Objekt an

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Bucket, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.
3. Wählen Sie in der Liste Objekte den Objektnamen aus.
4. Wählen Sie im Tab Details unter Serverseitige Verschlüsselungseinstellungen die Option Bearbeiten.

Unter Bucket-Schlüssel sehen Sie die Einstellungen des S3-Bucket-Schlüssels für Ihr Objekt. Sie können diese Einstellungen nicht bearbeiten.

Verwendung der AWS CLI

So zeigen Sie die S3-Bucket-Schlüssel-Einstellungen auf Bucket-Ebene an

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Weitere Informationen finden Sie [get-bucket-encryption](#) in der AWS CLI Befehlsreferenz.

So zeigen Sie die Einstellungen für einen S3-Bucket-Schlüssel auf Objektebene an

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

```
aws s3api head-object --bucket amzn-s3-demo-bucket1 --key my_images.tar.bz2
```

Weitere Informationen finden Sie unter [head-object](#) im AWS CLI Befehlsreferenz.

Mit dem REST API

So zeigen Sie die S3-Bucket-Schlüssel-Einstellungen auf Bucket-Ebene an

Verwenden Sie die Operation `GetBucketEncryption`, um Verschlüsselungsinformationen für einen Bucket anzuzeigen, einschließlich der Einstellungen für einen S3-Bucket-

Schlüssel. Die Einstellungen des S3-Bucket-Schlüssels werden im Antworttext im Element `ServerSideEncryptionConfiguration` mit der Einstellung `BucketKeyEnabled` angezeigt. Weitere Informationen finden Sie [GetBucketEncryption](#) in der Amazon S3 API S3-Referenz.

So zeigen Sie die Einstellungen für einen S3-Bucket-Schlüssel auf Objektebene an

Verwenden Sie den Vorgang `HeadObject`, um den S3-Bucket-Schlüssel-Status für ein Objekt anzuzeigen. `HeadObject` liefert den Antwort-Header `x-amz-server-side-encryption-bucket-key-enabled`, der zeigt, ob ein S3-Bucket-Schlüssel für das Objekt aktiviert oder deaktiviert ist. Weitere Informationen finden Sie [HeadObject](#) in der Amazon S3 API S3-Referenz.

Die folgenden API Operationen geben auch den `x-amz-server-side-encryption-bucket-key-enabled` Antwort-Header zurück, wenn ein S3-Bucket-Key für ein Objekt konfiguriert ist:


- [PutObject](#)
- [PostObject](#)
- [CopyObject](#)
- [CreateMultipartUpload](#)
- [UploadPartCopy](#)
- [UploadPart](#)
- [CompleteMultipartUpload](#)
- [GetObject](#)

Verwendung einer zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüssel (-) DSSE KMS

Verwendung einer zweischichtigen serverseitigen Verschlüsselung mit AWS Key Management Service (AWS KMS) keys (DSSE-KMS) wendet zwei Verschlüsselungsebenen auf Objekte an, wenn sie auf Amazon S3 hochgeladen werden. DSSE— KMS hilft Ihnen, Compliance-Standards, die eine mehrschichtige Verschlüsselung Ihrer Daten erfordern, einfacher zu erfüllen und die volle Kontrolle über Ihre Verschlüsselungsschlüssel zu haben.

Wenn Sie DSSE - KMS mit einem Amazon S3 S3-Bucket - verwenden AWS KMS Die Schlüssel müssen sich in derselben Region wie der Bucket befinden. Wenn DSSE - für das Objekt angefordert KMS wird, wird außerdem die S3-Prüfsumme, die Teil der Metadaten des Objekts ist, in verschlüsselter Form gespeichert. Weitere Informationen zu Prüfsummen finden Sie unter [Überprüfung der Objektintegrität](#).

Für die Nutzung von DSSE - KMS und fallen zusätzliche Gebühren an AWS KMS keys. Weitere Informationen zur DSSE KMS Preisgestaltung finden Sie unter [AWS KMS key Konzepte](#) in der AWS Key Management Service Entwicklerhandbuch und [AWS KMS Preisgestaltung](#).

 Note

S3-Bucket-Keys werden für DSSE - nicht unterstütztKMS.

Erfordert serverseitige Dual-Layer-Verschlüsselung mit AWS KMS keys (DSSE-KMS)

Wenn Sie die serverseitige Dual-Layer-Verschlüsselung aller Objekte in einem bestimmten Amazon-S3-Bucket anfordern möchten, können Sie eine Bucket-Richtlinie verwenden. Die folgende Bucket-Richtlinie verweigert beispielsweise allen Benutzern die Berechtigung Objekt (s3:PutObject) hochladen, wenn die Anfrage keinen x-amz-server-side-encryption Header enthält, der eine serverseitige Verschlüsselung mit - anfordert. DSSE KMS

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyUnEncryptedObjectUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "aws:kms:dsse"
        }
      }
    }
  ]
}
```

Themen

- [Spezifizierung der zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüssel \(-\) DSSE KMS](#)

Spezifizierung der zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüssel (-) DSSE KMS

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit verwenden AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder SSE - KMS verwenden. DSSE KMS

Sie können die Verschlüsselung anwenden, wenn Sie entweder ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren.

Sie können Folgendes DSSE angeben: KMS mithilfe der Amazon S3-Konsole RESTAPI, Amazon S3 und AWS Command Line Interface (AWS CLI). Weitere Informationen finden Sie in den folgenden Themen.

Note

Sie können Multiregion verwenden AWS KMS keys auf Amazon S3. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter [Verwenden von Schlüsseln für mehrere Regionen in AWS Key Management Service Entwicklerhandbuch](#).

Note

Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie die Erlaubnis, den Schlüssel zu verwenden. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [Erstellen von KMS Schlüsseln, die andere Konten verwenden können](#), in AWS Key Management Service Leitfaden für Entwickler.

Verwenden der S3-Konsole

In diesem Abschnitt wird beschrieben, wie Sie den Verschlüsselungstyp eines Objekts für die serverseitige Dual-Layer-Verschlüsselung festlegen oder ändern AWS Key Management Service (AWS KMS) Tasten (DSSE-KMS) mithilfe der Amazon S3 S3-Konsole.

Note

- Wenn Sie die Verschlüsselungsmethode eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).
- Wenn Sie den Verschlüsselungstyp für ein Objekt mit benutzerdefinierten Tags ändern, benötigen Sie die entsprechende `s3:GetObjectTagging` Genehmigung. Wenn Sie den Verschlüsselungstyp für ein Objekt ändern, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Genehmigung.

Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, wird der Verschlüsselungstyp für das Objekt aktualisiert, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

So fügen Sie die Verschlüsselung für ein Objekt hinzu oder ändern Sie sie

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält, das Sie verschlüsseln möchten.
4. Wählen Sie in der Liste Objekte das Kontrollkästchen neben dem Objekt aus, für das Sie eine Verschlüsselung hinzufügen oder ändern möchten.

Die Detailseite des Objekts wird angezeigt. Sie enthält mehrere Abschnitte mit den Eigenschaften des Objekts.

5. Wählen Sie die Registerkarte Eigenschaften aus.
6. Scrollen Sie nach unten zum Abschnitt Standardverschlüsselung und wählen Sie Bearbeiten aus.

Die Seite Standardverschlüsselung bearbeiten wird geöffnet.

7. Wählen Sie unter Verschlüsselungstyp die Option Serverseitige Dual-Layer-Verschlüsselung mit AWS Key Management Service Schlüssel (DSSE-KMS).
8. Unter AWS KMS Taste, führen Sie einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie dann Ihren KMS Schlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (`aws/s3`) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie dann Ihren KMS Schlüssel ARN in das angezeigte Feld ein.

- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

Important

Sie können nur KMS Schlüssel verwenden, die in derselben Version verfügbar sind AWS-Region wie der Eimer. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN.

Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter [Identifizieren asymmetrischer Schlüssel KMS](#) in der AWS Key Management Service Leitfaden für Entwickler.

9. Wählen Sie unter Bucket-Schlüssel die Option Deaktivieren aus. S3-Bucket-Keys werden für DSSE - nicht unterstütztKMS.
10. Wählen Sie Änderungen speichern.

Note

Diese Aktion wendet auf alle angegebenen Objekte Verschlüsselung an. Warten Sie beim Verschlüsseln von Ordnern, bis die Speicheroperation abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen.

Mit dem REST API

Wenn Sie ein Objekt erstellen, d. h. wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die Verwendung der zweischichtigen serverseitigen Verschlüsselung mit angeben AWS KMS keys (-) um Ihre Daten zu verschlüsseln. DSSE KMS Fügen Sie hierzu der

Anforderung den Header `x-amz-server-side-encryption` hinzu. Setzen Sie den Wert des Headers auf den `aws:kms:dsse`-Verschlüsselungsalgorithmus. Amazon S3 bestätigt, dass Ihr Objekt KMS verschlüsselt gespeichert wurde, indem es den Antwort-Header zurückgibt `x-amz-server-side-encryption: DSSE`

Wenn Sie den Header `x-amz-server-side-encryption` mit dem Wert `aws:kms:dsse` angeben, können Sie auch die folgenden Anforderungs-Header verwenden:

- `x-amz-server-side-encryption-aws-kms-key-id`: *SSEKMSKeyId*
- `x-amz-server-side-encryption-context`: *SSEKMSEncryptionContext*

Themen

- [Amazon S3 REST API S3-Operationen, die unterstützen DSSE - KMS](#)
- [Verschlüsselungskontext \(x-amz-server-side-encryption-context\)](#)
- [AWS KMS Schlüssel-ID \(x-amz-server-side-encryption-aws-kms-key-id\)](#)


Amazon S3 REST API S3-Operationen, die unterstützen DSSE - KMS

Die folgenden REST API Operationen akzeptieren die Header `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id`, und `x-amz-server-side-encryption-context` Request-Header.

- [PutObject](#)— Wenn Sie Daten mithilfe der PUT API Operation hochladen, können Sie diese Anforderungsheader angeben.
- [CopyObject](#) – Wenn Sie ein Objekt kopieren, erhalten Sie ein Quell- und ein Zielobjekt. Wenn Sie bei der CopyObject Operation DSSE KMS -Header übergeben, werden sie nur auf das Zielobjekt angewendet. Beim Kopieren eines vorhandenen Objekts wird das Zielobjekt unabhängig davon, ob das Quellobjekt verschlüsselt ist, nur dann verschlüsselt, wenn Sie die serverseitige Verschlüsselung explizit anfordern.
- [POST Objekt](#) — Wenn Sie eine POST Operation verwenden, um ein Objekt hochzuladen, geben Sie anstelle der Anforderungsheader dieselben Informationen in die Formularfelder ein.
- [CreateMultipartUpload](#) – Wenn Sie große Objekte über einen mehrteiligen Upload hochladen, können Sie diese Header in der CreateMultipartUpload-Anforderung angeben.

Die Antwortheader der folgenden REST API Operationen geben den `x-amz-server-side-encryption` Header zurück, wenn ein Objekt mit serverseitiger Verschlüsselung gespeichert wird.

- [PutObject](#)
- [CopyObject](#)
- [POST Objekt](#)
- [CreateMultipartUpload](#)
- [UploadPart](#)
- [UploadPartCopy](#)
- [CompleteMultipartUpload](#)
- [GetObject](#)
- [HeadObject](#)

 **Important**

- Alle GET und PUT Anfragen für ein Objekt, das geschützt ist durch AWS KMS schlagen fehl, wenn Sie sie nicht mithilfe von Secure Sockets Layer (SSL), Transport Layer Security (TLS) oder Signature Version 4 erstellen.
- Wenn Ihr Objekt DSSE - verwendet KMS, senden Sie keine Header für GET Verschlüsselungsanfragen für Anfragen und HEAD Anfragen, da Sie sonst die Fehlermeldung HTTP400 (Bad Request) erhalten.

Verschlüsselungskontext (**x-amz-server-side-encryption-context**)

Wenn Sie angeben `x-amz-server-side-encryption:aws:kms:dsse`, API unterstützt Amazon S3 einen Verschlüsselungskontext mit dem `x-amz-server-side-encryption-context` Header. Ein Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten können.

Amazon S3 verwendet automatisch den Amazon-Ressourcennamen (ARN) des Objekts als Verschlüsselungskontextpaar; zum Beispiel `arn:aws:s3:::object_ARN`.

Sie können optional ein zusätzliches Verschlüsselungskontextpaar bereitstellen, indem Sie den Header `x-amz-server-side-encryption-context` verwenden. Da der Verschlüsselungskontext jedoch nicht verschlüsselt ist, sollte er keine sensiblen Informationen enthalten. Amazon S3 speichert dieses zusätzliche Schlüsselpaar zusammen mit dem Standardverschlüsselungskontext.

Weitere Informationen zum Verschlüsselungskontext in Amazon S3 finden Sie unter [Verschlüsselungskontext](#). Allgemeine Informationen zum Verschlüsselungskontext finden Sie unter [AWS Key Management Service Konzepte — Verschlüsselungskontext](#) im AWS Key Management Service Leitfaden für Entwickler.

AWS KMS Schlüssel-ID (**x-amz-server-side-encryption-aws-kms-key-id**)

Sie können den Header `x-amz-server-side-encryption-aws-kms-key-id` verwenden, um die ID des vom Kunden verwalteten Schlüssels anzugeben, der zum Schutz der Daten verwendet wird. Wenn Sie den `x-amz-server-side-encryption:aws:kms:dsse` Header angeben, den `x-amz-server-side-encryption-aws-kms-key-id` Header aber nicht angeben, verwendet Amazon S3 den Von AWS verwalteter Schlüssel (`aws/s3`) um die Daten zu schützen. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden möchten, müssen Sie den `x-amz-server-side-encryption-aws-kms-key-id`-Header des vom Kunden verwalteten Schlüssels angeben.

Important

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter [Symmetrische KMS Verschlüsselungsschlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Verwendung der AWS CLI

Wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die Verwendung von DSSE - KMS zur Verschlüsselung Ihrer Daten angeben. Fügen Sie hierzu der Anforderung den Parameter `--server-side-encryption aws:kms:dsse` hinzu. Verwenden Sie den `--ssekms-key-id` *example-key-id* Parameter, um Ihren [Kunden hinzuzufügen \(verwaltet\) AWS KMS Schlüssel](#), den Sie erstellt haben. Wenn Sie einen angeben `--server-side-encryption aws:kms:dsse`, aber nicht angeben AWS KMS Schlüssel-ID, dann verwendet Amazon S3 die AWS verwalteter Schlüssel (`aws/s3`).

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --server-side-encryption aws:kms:dsse --ssekms-key-id example-key-id --body filepath
```

Sie können ein unverschlüsseltes Objekt verschlüsseln, um es zu verwenden, KMS indem DSSE Sie das Objekt wieder an seinen ursprünglichen Platz kopieren.

```
aws s3api copy-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --  
body filepath --bucket DOC-EXAMPLE-BUCKET --key example-object-key --sse aws:kms:dsse  
--sse-kms-key-id example-key-id --body filepath
```

Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Die serverseitige Verschlüsselung dient zum Schutz ruhender Daten. Die serverseitige Verschlüsselung verschlüsselt nur die Objektdaten, nicht die Metadaten des Objekts. Durch die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) können Sie Ihre Daten mit Ihren eigenen Verschlüsselungsschlüsseln verschlüsselt speichern. Mit dem Verschlüsselungsschlüssel, den Sie als Teil Ihrer Anforderung bereitstellen, verwaltet Amazon S3 die Datenverschlüsselung, wenn es auf Datenträger schreibt, und die Entschlüsselung, wenn Sie auf Ihre Objekte zugreifen. Sie müssen also für die Datenverschlüsselung und -entschlüsselung keinen Code mehr verwalten. Sie müssen nur noch die von Ihnen bereitgestellten Verschlüsselungsschlüssel verwalten.

Wenn Sie ein Objekt hochladen, verwendet Amazon S3 den von Ihnen angegebenen Verschlüsselungsschlüssel, um die AES 256-Verschlüsselung auf Ihre Daten anzuwenden. Amazon S3 entfernt dann den Verschlüsselungsschlüssel aus dem Speicher. Wenn Sie ein Objekt abrufen, müssen Sie denselben Verschlüsselungsschlüssel als Teil Ihrer Anfrage angeben. Amazon S3 überprüft zuerst, ob der von Ihnen bereitgestellte Verschlüsselungsschlüssel übereinstimmt, und entschlüsselt das Objekt, bevor Objektdaten zurückgegeben werden.

Für die Verwendung von SSE -C fallen keine zusätzlichen Gebühren an. Für Anfragen zur Konfiguration und Verwendung von SSE -C fallen jedoch Standardgebühren für Amazon S3 S3-Anfragen an. Informationen zu Preisen finden Sie unter [Amazon S3 – Preise](#).

Note

Amazon S3 speichert den von Ihnen bereitgestellten Verschlüsselungsschlüssel nicht. Stattdessen wird ein zufällig gesalzener Hash-basierter Message Authentication Code (HMAC) -Wert des Verschlüsselungsschlüssels gespeichert, um future Anfragen zu validieren. Der gesalzene HMAC Wert kann nicht verwendet werden, um den Wert des Verschlüsselungsschlüssels abzuleiten oder den Inhalt des verschlüsselten Objekts zu entschlüsseln. Das bedeutet, wenn Sie den Verschlüsselungsschlüssel verlieren, verlieren Sie das Objekt.

Die S3-Replikation unterstützt Objekte, die mit SSE -C verschlüsselt sind. Weitere Hinweise zur Replikation verschlüsselter Objekte finden Sie unter. [the section called “Replizieren verschlüsselter Objekte”](#)

Weitere Informationen zu SSE -C finden Sie in den folgenden Themen.


Themen

- [SSE-C im Überblick](#)
- [-C erforderlich und einschränkend SSE](#)
- [Vorsigniert und URLs -C SSE](#)
- [Angaben der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln \(-C\) SSE](#)

SSE-C im Überblick

Dieser Abschnitt bietet einen Überblick über SSE -C. Beachten Sie bei der Verwendung von SSE -C die folgenden Überlegungen.


- Sie müssen HTTPS verwenden.

 **Important**

Amazon S3 lehnt alle Anfragen ab, die HTTP bei Verwendung von SSE -C gestellt werden. Aus Sicherheitsgründen empfehlen wir, dass Sie jeden Schlüssel, den Sie irrtümlich versenden, als HTTP kompromittiert betrachten. Verwerfen Sie den Schlüssel und rotieren Sie ihn wie erforderlich.

- Das Entity-Tag (ETag) in der Antwort ist nicht der MD5 Hash der Objektdaten.
- Sie Verwalten ein Mapping, welcher Verschlüsselungsschlüssel für die Verschlüsselung welches Objekts verwendet wurde. Amazon S3 speichert keine Verschlüsselungsschlüssel. Sie sind dafür verantwortlich, zu verwalten, welchen Verschlüsselungsschlüssel Sie für welches Objekt angegeben haben.
- Wenn für Ihren Bucket Versioning aktiviert ist, kann jede Objektversion, die Sie mit dieser Funktion hochladen, einen eigenen Verschlüsselungsschlüssel haben. Sie sind dafür verantwortlich, zu verwalten, welcher Verschlüsselungsschlüssel für welche Objektversion verwendet wurde.

- Sie verwalten die Verschlüsselungsschlüssel auf der Clientseite, deshalb verwalten Sie auch alle zusätzlichen Sicherungsmechanismen auf der Clientseite, wie beispielsweise die Schlüsselrotation.

 Warning

Wenn Sie den Verschlüsselungsschlüssel verlieren, schlagen alle GET-Anfragen für ein Objekt ohne seinen Verschlüsselungsschlüssel fehl und Sie verlieren das Objekt.

-C erforderlich und einschränkend SSE

Um SSE -C für alle Objekte in einem bestimmten Amazon S3 S3-Bucket zu verlangen, können Sie eine Bucket-Richtlinie verwenden.

Die folgende Bucket-Richtlinie verweigert beispielsweise Upload object (s3:PutObject) - Berechtigungen für alle Anfragen, die nicht den x-amz-server-side-encryption-customer-algorithm Header enthalten, der -C anfordert SSE.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "RequireSSECObjectUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption-customer-algorithm": "true"
        }
      }
    }
  ]
}
```

Sie können auch eine Richtlinie verwenden, um die serverseitige Verschlüsselung aller Objekte in einem bestimmten Amazon-S3-Bucket einzuschränken. Die folgende Bucket-Richtlinie verweigert

beispielsweise allen Benutzern die Berechtigung zum Hochladen von Objekten (s3:PutObject), wenn die Anfrage den `x-amz-server-side-encryption-customer-algorithm` Header enthält, der `-C` anfordert. SSE

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "RestrictSSEObjectUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption-customer-algorithm": "false"
        }
      }
    }
  ]
}
```

Important

Wenn Sie eine Bucket-Richtlinie verwenden, bei der die Option SSE `-C` aktiviert werden muss `s3:PutObject`, müssen Sie den `x-amz-server-side-encryption-customer-algorithm` Header in alle mehrteiligen Upload-Anfragen (`CreateMultipartUpload`, `UploadPart` und) aufnehmen. `CompleteMultipartUpload`

Vorsigniert und URLs `-C` SSE

Sie können ein vorsigniertes Objekt generieren URL, das für Operationen wie das Hochladen eines neuen Objekts, das Abrufen eines vorhandenen Objekts oder das Abrufen von Objektmetadaten verwendet werden kann. Vorsignierte unterstützen URLs `-C` wie folgt: SSE

- Wenn Sie eine vorsignierte Datei erstellen URL, müssen Sie den Algorithmus angeben, indem Sie den `x-amz-server-side-encryption-customer-algorithm` Header in der Signaturberechnung verwenden.

- Wenn Sie Presigned verwenden, URL um ein neues Objekt hochzuladen, ein vorhandenes Objekt abzurufen oder nur Objektmetadaten abzurufen, müssen Sie alle Verschlüsselungsheader in der Anfrage Ihrer Client-Anwendung angeben.

Note

Für Objekte, die nicht zu SSE C gehören, können Sie ein vorsigniertes Objekt generieren URL und dieses direkt URL in einen Browser einfügen, um auf die Daten zuzugreifen. Bei SSE -C-Objekten ist dies jedoch nicht möglich, da Sie zusätzlich zu den vorsignierten Objekten auch HTTP Header angeben müssen URL, die für -C-Objekte spezifisch sind. SSE Daher können Sie vorsignierte Objekte nur programmgesteuert URLs für SSE -C-Objekte verwenden.

Weitere Hinweise zu presigned finden Sie unter. URLs [the section called “Arbeitet mit presigned URLs”](#)

Angeben der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Bei der Objekterstellung mit dem können Sie die REST API serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) angeben. Wenn Sie SSE -C verwenden, müssen Sie die Informationen zum Verschlüsselungsschlüssel mithilfe der folgenden Anforderungsheader angeben.

Name	Beschreibung
x-amz-server-side-encryption-customer-algorithm	Verwenden Sie diesen Header, um den Verschlüsselungsalgorithmus anzugeben. Der Header-Wert muss AES256 sein.
x-amz-server-side-encryption-customer-key	Verwenden Sie diesen Header, um den base64-codierten 256-bit-Verschlüsselungsschlüssel für Amazon S3 bereitzustellen, mit dem Ihre Daten verschlüsselt und entschlüsselt werden.
x-amz-server-side-encryption-customer-key-MD5	Verwenden Sie diesen Header, um den Base64-codierten MD5 128-Bit-Digest des Verschlüsselungsschlüssels gemäß 1321 bereitzustellen. RFC Amazon S3 verwendet diesen Header für eine Überprüfung

Name	Beschreibung
	ng der Nachrichtenintegrität, um sicherzustellen, dass der Verschlüsselungsschlüssel fehlerfrei übertragen wurde.

Sie können Folgendes verwenden ... AWS SDKWrapper-Bibliotheken, um diese Header zu Ihrer Anfrage hinzuzufügen. Bei Bedarf können Sie die Amazon S3 REST API S3-Aufrufe direkt in Ihrer Anwendung tätigen.

Note

Sie können die Amazon S3 S3-Konsole nicht verwenden, um ein Objekt hochzuladen und SSE -C anzufordern. Sie können die Konsole auch nicht verwenden, um ein vorhandenes Objekt, das mit SSE -C gespeichert wurde, zu aktualisieren (z. B. die Speicherklasse zu ändern oder Metadaten hinzuzufügen).

Verwenden von REST API

Amazon S3 S3-RestAPIs, der SSE -C unterstützt

Die folgenden Amazon S3 APIs unterstützen serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C).

- GETVorgang — Beim Abrufen von Objekten mithilfe von GET API (siehe [GETObjekt](#)) können Sie die Anforderungsheader angeben.
- HEADOperation — Um Objektmetadaten mithilfe von HEAD API (siehe [HEADObjekt](#)) abzurufen, können Sie diese Anforderungsheader angeben.
- PUTOperation — Beim Hochladen von Daten mithilfe des PUT Objekts API (siehe [PUTObjekt](#)) können Sie diese Anforderungsheader angeben.
- Mehrteiliger Upload — Wenn Sie große Objekte mithilfe des mehrteiligen Uploads hochladenAPI, können Sie diese Header angeben. Sie geben diese Header in der Initiierungsanforderung (siehe [Initiieren mehrteiliger Uploads](#)) und in jeder nachfolgenden Anforderung für einen Teil-Upload an (siehe [Hochladen von Teilen](#) oder [Hochladen von Teilen – Kopieren](#)). Für jede teilweise Upload-Anfrage muss dieselbe Verschlüsselungsinformation angegeben werden, wie diejenige, die Sie in der Initiierungsanfrage für den mehrteiligen Upload angegeben haben.

- **POSTOperation** — Wenn Sie einen POST Vorgang zum Hochladen eines Objekts (siehe [POSTObjekt](#)) verwenden, geben Sie anstelle der Anforderungsheader dieselben Informationen in die Formularfelder ein.
- **Kopiervorgang** — Wenn Sie ein Objekt kopieren (siehe [PUTObjekt — Kopieren](#)), haben Sie sowohl ein Quell- als auch ein Zielobjekt:
 - Wenn Sie möchten, dass das Zielobjekt serverseitig verschlüsselt wird mit AWS Bei verwalteten Schlüsseln müssen Sie den `x-amz-server-side-encryption` Anforderungsheader angeben.
 - Wenn Sie möchten, dass das Zielobjekt mit SSE -C verschlüsselt wird, müssen Sie Verschlüsselungsinformationen mithilfe der drei in der vorherigen Tabelle beschriebenen Header angeben.
 - Wenn das Quellobjekt mit SSE -C verschlüsselt ist, müssen Sie die Informationen zum Verschlüsselungsschlüssel mithilfe der folgenden Header angeben, damit Amazon S3 das Objekt zum Kopieren entschlüsseln kann.

Name	Beschreibung
<code>x-amz-copy-source-server-side-encryption-customer-algorithm</code>	Verwenden Sie diesen Header, um den Verschlüsselungsalgorithmus anzugeben, den Amazon S3 für die Entschlüsselung des Quellobjekts verwenden soll. Dieser Wert muss AES256 lauten.
<code>x-amz-copy-source-server-side-encryption-customer-key</code>	Verwenden Sie diesen Header, um den base64-codierten 256-bit-Verschlüsselungsschlüssel für Amazon S3 bereitzustellen, mit dem das Quellobjekt entschlüsselt werden soll. Dieser Verschlüsselungsschlüssel muss derjenige sein, den Sie Amazon S3 beim Erstellen des Quellobjekts bereitgestellt haben. Andernfalls kann Amazon S3 das Objekt nicht entschlüsseln.
<code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code>	Fügen Sie diesen Header hinzu, um den Base64-codierten MD5 128-Bit-Digest des Verschlüsselungsschlüssels gemäß 1321 bereitzustellen. RFC

Verwendung der AWS SDKsum SSE -C für PUT die Operationen,, Head und Copy anzugeben GET

Die folgenden Beispiele zeigen, wie eine serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) für Objekte angefordert wird. Dabei werden folgende Vorgänge ausgeführt. Jeder Vorgang zeigt, wie Header im SSE Zusammenhang mit -C in der Anfrage angegeben werden:

- Put object – Lädt ein Objekt hoch und fordert die serverseitige Verschlüsselung mit einem vom Kunden bereitgestellten Verschlüsselungsschlüssel an.
- Get object – Lädt das im vorigen Schritt hochgeladene Objekt herunter. Sie stellen in der Anforderung dieselben Verschlüsselungs-Informationen bereit, die Sie beim Hochladen des Objekts angegeben haben. Amazon S3 benötigt diese Informationen, um das Objekt zu entschlüsseln und an Sie zurückzugeben.
- Get object metadata – Ruft die Metadaten des Objekts ab. Sie stellen dieselben Verschlüsselungs-Informationen bereit, die beim Erstellen des Objekts verwendet wurden.
- Copy object – Erstellt eine Kopie des zuvor hochgeladenen Objekts. Da das Quellobjekt mit SSE - C gespeichert wird, müssen Sie seine Verschlüsselungsinformationen in Ihrer Kopieranforderung angeben. Standardmäßig verschlüsselt Amazon S3 die Kopie des Objekts nur, wenn Sie dies ausdrücklich anfordern. In diesem Beispiel wird Amazon S3 dazu angewiesen, eine verschlüsselte Kopie des Objekts zu speichern.

Java

Note

Dieses Beispiel zeigt, wie ein Objekt in einer einzigen Operation hochgeladen wird. Wenn Sie den Multipart Upload verwenden, API um große Objekte hochzuladen, geben Sie die Verschlüsselungsinformationen auf die gleiche Weise ein, wie in diesem Beispiel gezeigt. Beispiele für mehrteilige Uploads, die den AWS SDK for Java, finden Sie unter [Hochladen eines Objekts mit Multipart-Upload](#).

Um die erforderlichen Verschlüsselungsinformationen hinzuzufügen, schließen Sie in Ihre Anfrage einen `SSECustomerKey` ein. Weitere Informationen zu der `SSECustomerKey` Klasse finden Sie im REST API Abschnitt.

Hinweise zu SSE -C finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln \(-C\) SSE](#). Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import javax.crypto.KeyGenerator;
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

public class ServerSideEncryptionUsingClientSideEncryptionKey {
    private static SSECustomerKey SSE_KEY;
    private static AmazonS3 S3_CLIENT;
    private static KeyGenerator KEY_GENERATOR;

    public static void main(String[] args) throws IOException,
        NoSuchAlgorithmException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";
        String uploadFileName = "**** File path ****";
        String targetKeyName = "**** Target key name ****";

        // Create an encryption key.
        KEY_GENERATOR = KeyGenerator.getInstance("AES");
        KEY_GENERATOR.init(256, new SecureRandom());
        SSE_KEY = new SSECustomerKey(KEY_GENERATOR.generateKey());

        try {
```

```
S3_CLIENT = AmazonS3ClientBuilder.standard()
    .withCredentials(new ProfileCredentialsProvider())
    .withRegion(clientRegion)
    .build();

// Upload an object.
uploadObject(bucketName, keyName, new File(uploadFileName));

// Download the object.
downloadObject(bucketName, keyName);

// Verify that the object is properly encrypted by attempting to
retrieve it
// using the encryption key.
retrieveObjectMetadata(bucketName, keyName);

// Copy the object into a new object that also uses SSE-C.
copyObject(bucketName, keyName, targetKeyName);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

private static void uploadObject(String bucketName, String keyName, File file) {
    PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
file).withSSECustomerKey(SSE_KEY);
    S3_CLIENT.putObject(putRequest);
    System.out.println("Object uploaded");
}

private static void downloadObject(String bucketName, String keyName) throws
IOException {
    GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName,
keyName).withSSECustomerKey(SSE_KEY);
    S3Object object = S3_CLIENT.getObject(getObjectRequest);

    System.out.println("Object content: ");
    displayTextInputStream(object.getObjectContent());
}
```



```
    }

    private static void retrieveObjectMetadata(String bucketName, String keyName) {
        GetObjectMetadataRequest getMetadataRequest = new
GetObjectMetadataRequest(bucketName, keyName)
            .withSSECustomerKey(SSE_KEY);
        ObjectMetadata objectMetadata =
S3_CLIENT.getObjectMetadata(getMetadataRequest);
        System.out.println("Metadata retrieved. Object size: " +
objectMetadata.getContentLength());
    }

    private static void copyObject(String bucketName, String keyName, String
targetKeyName)
        throws NoSuchAlgorithmException {
        // Create a new encryption key for target so that the target is saved using
// SSE-C.
        SSECustomerKey newSSEKey = new SSECustomerKey(KEY_GENERATOR.generateKey());

        CopyObjectRequest copyRequest = new CopyObjectRequest(bucketName, keyName,
bucketName, targetKeyName)
            .withSourceSSECustomerKey(SSE_KEY)
            .withDestinationSSECustomerKey(newSSEKey);

        S3_CLIENT.copyObject(copyRequest);
        System.out.println("Object copied");
    }

    private static void displayTextInputStream(S3ObjectInputStream input) throws
IOException {
        // Read one line at a time from the input stream and display each line.
        BufferedReader reader = new BufferedReader(new InputStreamReader(input));
        String line;
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
        System.out.println();
    }
}
```

.NET

Note

Beispiele für das Hochladen großer Objekte mithilfe des mehrteiligen Uploads finden Sie unter API [Hochladen eines Objekts mit Multipart-Upload](#) und [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#)

Hinweise zu SSE -C finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln \(-C\) SSE](#) Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSEClientEncryptionKeyObjectOperationsTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string keyName = "**** key name for new object created ****";
        private const string copyTargetKeyName = "**** key name for object copy ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            ObjectOpsUsingClientEncryptionKeyAsync().Wait();
        }
        private static async Task ObjectOpsUsingClientEncryptionKeyAsync()
```

```
{
    try
    {
        // Create an encryption key.
        Aes aesEncryption = Aes.Create();
        aesEncryption.KeySize = 256;
        aesEncryption.GenerateKey();
        string base64Key = Convert.ToBase64String(aesEncryption.Key);

        // 1. Upload the object.
        PutObjectRequest putObjectRequest = await
UploadObjectAsync(base64Key);
        // 2. Download the object and verify that its contents matches what
you uploaded.
        await DownloadObjectAsync(base64Key, putObjectRequest);
        // 3. Get object metadata and verify that the object uses AES-256
encryption.
        await GetObjectMetadataAsync(base64Key);
        // 4. Copy both the source and target objects using server-side
encryption with
        //    a customer-provided encryption key.
        await CopyObjectAsync(aesEncryption, base64Key);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}

private static async Task<PutObjectRequest> UploadObjectAsync(string
base64Key)
{
    PutObjectRequest putObjectRequest = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = keyName,
        ContentBody = "sample text",
```

```
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };
    PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
    return putObjectRequest;
}
private static async Task DownloadObjectAsync(string base64Key,
PutObjectRequest putObjectRequest)
{
    GetObjectRequest getObjectRequest = new GetObjectRequest
    {
        BucketName = bucketName,
        Key = keyName,
        // Provide encryption information for the object stored in Amazon
S3.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };

    using (GetObjectResponse getResponse = await
client.GetObjectAsync(getObjectRequest))
        using (StreamReader reader = new
StreamReader(getResponse.ResponseStream))
        {
            string content = reader.ReadToEnd();
            if (String.Compare(putObjectRequest.ContentBody, content) == 0)
                Console.WriteLine("Object content is same as we uploaded");
            else
                Console.WriteLine("Error...Object content is not same.");

            if (getResponse.ServerSideEncryptionCustomerMethod ==
ServerSideEncryptionCustomerMethod.AES256)
                Console.WriteLine("Object encryption method is AES256, same as
we set");
            else
                Console.WriteLine("Error...Object encryption method is not the
same as AES256 we set");

            // Assert.AreEqual(putObjectRequest.ContentBody, content);
            // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getResponse.ServerSideEncryptionCustomerMethod);
        }
    }
}
```

```
    }
}
private static async Task GetObjectMetadataAsync(string base64Key)
{
    GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest
    {
        BucketName = bucketName,
        Key = keyName,

        // The object stored in Amazon S3 is encrypted, so provide the
necessary encryption information.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };

    GetObjectMetadataResponse getObjectMetadataResponse = await
client.GetObjectMetadataAsync(getObjectMetadataRequest);
    Console.WriteLine("The object metadata show encryption method used is:
{0}", getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
    // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
}
private static async Task CopyObjectAsync(Aes aesEncryption, string
base64Key)
{
    aesEncryption.GenerateKey();
    string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);

    CopyObjectRequest copyRequest = new CopyObjectRequest
    {
        SourceBucket = bucketName,
        SourceKey = keyName,
        DestinationBucket = bucketName,
        DestinationKey = copyTargetKeyName,
        // Information about the source object's encryption.
        CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,
        // Information about the target object's encryption.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = copyBase64Key
    }
}
```

```
        };  
        await client.CopyObjectAsync(copyRequest);  
    }  
}  
}
```

Verwendung der AWS SDKsum SSE -C für mehrteilige Uploads anzugeben

Das Beispiel im vorherigen Abschnitt zeigt, wie Sie bei den Vorgängen, Head und Copy eine serverseitige Verschlüsselung mit einem vom Kunden bereitgestellten Schlüssel (SSE-C) anfordern. PUT GET In diesem Abschnitt werden andere Amazon S3 beschriebenen APIs, die SSE -C unterstützen.

Java

Um große Objekte hochzuladen, können Sie den mehrteiligen Upload verwenden API (siehe [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#)). APIs Um große Objekte hochzuladen, können Sie entweder High-Level oder Low-Level verwenden. Diese APIs unterstützen verschlüsselungsbezogene Header in der Anfrage.

- Wenn Sie den High-Level verwenden `TransferManager` API, geben Sie die verschlüsselungsspezifischen Header in der an (siehe). `PutObjectRequest` [Hochladen eines Objekts mit Multipart-Upload](#)
- Wenn Sie den Low-Level-Modus verwenden API, geben Sie verschlüsselungsbezogene Informationen in den `initiateMultipartUploadRequest`, gefolgt von identischen Verschlüsselungsinformationen in jedem. `UploadPartRequest` In Ihrer `CompleteMultipartUploadRequest` müssen Sie keine verschlüsselungsspezifischen Header angeben. Beispiele finden Sie unter [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#).

Das folgende Beispiel dient der Erstellung von Objekten und zeigt `TransferManager`, wie Informationen im Zusammenhang mit SSE -C bereitgestellt werden. Das Beispiel erledigt Folgendes:

- Erstellt ein Objekts mit der Methode `TransferManager.upload()`. In der Instance `PutObjectRequest` geben Sie anzufordernde Verschlüsselungsschlüssel-Informationen an. Amazon S3 verschlüsselt das Objekt mit dem vom Kunden bereitgestellten Schlüssel.
- Erstellt eine Kopie des Objekts durch Aufrufen der `TransferManager.copy()`-Methode. Das Beispiel weist Amazon S3 dazu an, eine Objektkopie unter Verwendung eines neuen

SSECustomerKey zu verschlüsseln. Da das Quellobjekt mit SSE -C verschlüsselt ist, stellt der CopyObjectRequest auch den Verschlüsselungsschlüssel des Quellobjekts bereit, sodass Amazon S3 das Objekt vor dem Kopieren entschlüsseln kann.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.SSECustomerKey;
import com.amazonaws.services.s3.transfer.Copy;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

import javax.crypto.KeyGenerator;
import java.io.File;
import java.security.SecureRandom;

public class ServerSideEncryptionCopyObjectUsingHLwithSSEC {

    public static void main(String[] args) throws Exception {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String fileToUpload = "**** File path ****";
        String keyName = "**** New object key name ****";
        String targetKeyName = "**** Key name for object copy ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .build();

            TransferManager tm = TransferManagerBuilder.standard()
                .withS3Client(s3Client)
                .build();
```

```
        // Create an object from a file.
        PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName,
keyName, new File(fileToUpload));

        // Create an encryption key.
        KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
        keyGenerator.init(256, new SecureRandom());
        SSECustomerKey sseCustomerEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());

        // Upload the object. TransferManager uploads asynchronously, so this
call
        // returns immediately.
        putObjectRequest.setSSECustomerKey(sseCustomerEncryptionKey);
        Upload upload = tm.upload(putObjectRequest);

        // Optionally, wait for the upload to finish before continuing.
        upload.waitForCompletion();
        System.out.println("Object created.");

        // Copy the object and store the copy using SSE-C with a new key.
        CopyObjectRequest copyObjectRequest = new CopyObjectRequest(bucketName,
keyName, bucketName, targetKeyName);
        SSECustomerKey sseTargetObjectEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());
        copyObjectRequest.setSourceSSECustomerKey(sseCustomerEncryptionKey);

        copyObjectRequest.setDestinationSSECustomerKey(sseTargetObjectEncryptionKey);

        // Copy the object. TransferManager copies asynchronously, so this call
returns
        // immediately.
        Copy copy = tm.copy(copyObjectRequest);

        // Optionally, wait for the upload to finish before continuing.
        copy.waitForCompletion();
        System.out.println("Copy complete.");
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
```



```
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Um große Objekte hochzuladen, können Sie den mehrteiligen Upload verwenden API (siehe). [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#) AWS SDK für .NET ermöglicht das Hochladen großer Objekte sowohl auf hoher als auch APIs auf niedriger Ebene. Diese APIs unterstützen verschlüsselungsbezogene Header in der Anfrage.

- Wenn Sie High-Level verwenden Transfer-Utility API, geben Sie die verschlüsselungsspezifischen Header in der Abbildung an. TransferUtilityUploadRequest Codebeispiele finden Sie unter [Hochladen eines Objekts mit Multipart-Upload](#).

```
TransferUtilityUploadRequest request = new TransferUtilityUploadRequest()
{
    FilePath = filePath,
    BucketName = existingBucketName,
    Key = keyName,
    // Provide encryption information.
    ServerSideEncryptionCustomerMethod =
    ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key,
};
```

- Bei Verwendung der Low-Level-Methode API geben Sie Informationen zur Verschlüsselung in der Anfrage zur Initialisierung eines mehrteiligen Uploads ein, gefolgt von identischen Verschlüsselungsinformationen in den nachfolgenden Upload-Teilansforderungen. In der vollständigen Anforderung zum mehrteiligen Upload müssen Sie keine verschlüsselungsspezifischen Header angeben. Beispiele finden Sie unter [Verwendung der AWS SDKs\(niedriges NiveauAPI\)](#).

Nachfolgend finden Sie ein Beispiel für einen mehrteiligen Low-Level-Upload, der eine Kopie eines vorhandenen großen Objekts erstellt. In dem Beispiel wird das zu kopierende Objekt mit SSE -C in Amazon S3 gespeichert, und Sie möchten das Zielobjekt auch mit SSE -C speichern. In dem Beispiel gehen Sie wie folgt vor:

- Initiieren Sie eine mehrteilige Upload-Anfrage, indem Sie einen Verschlüsselungsschlüssel und zugehörige Informationen bereitstellen.
- Stellen Sie Verschlüsselungsschlüssel für das Quell- und das Zielobjekt sowie die zugehörigen Informationen in der CopyPartRequest bereit.
- Ermitteln Sie die Größe des zu kopierenden Quellobjekts, indem Sie die Objekt-Metadaten abrufen.
- Laden Sie die Objekte in Teilen mit je 5 MB hoch.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSECLowLevelMPUCopyObjectTest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string sourceKeyName     = "*** source object key name
***";
        private const string targetKeyName     = "*** key name for the target
object ***";
        private const string filePath         = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CopyObjClientEncryptionKeyAsync().Wait();
        }

        private static async Task CopyObjClientEncryptionKeyAsync()
        {
```

```
Aes aesEncryption = Aes.Create();
aesEncryption.KeySize = 256;
aesEncryption.GenerateKey();
string base64Key = Convert.ToBase64String(aesEncryption.Key);

await CreateSampleObjUsingClientEncryptionKeyAsync(base64Key,
s3Client);

await CopyObjectAsync(s3Client, base64Key);
}
private static async Task CopyObjectAsync(IAmazonS3 s3Client, string
base64Key)
{
    List<CopyPartResponse> uploadResponses = new List<CopyPartResponse>();

    // 1. Initialize.
    InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
    {
        BucketName = existingBucketName,
        Key = targetKeyName,
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key,
    };

    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initiateRequest);

    // 2. Upload Parts.
    long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB
    long firstByte = 0;
    long lastByte = partSize;

    try
    {
        // First find source object size. Because object is stored
encrypted with
        // customer provided key you need to provide encryption
information in your request.
        GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest()
        {
            BucketName = existingBucketName,
```

```
        Key = sourceKeyName,
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key // " *
**source object encryption key ***"
    };

    GetObjectMetadataResponse getObjectMetadataResponse = await
s3Client.GetObjectMetadataAsync(getObjectMetadataRequest);

    long filePosition = 0;
    for (int i = 1; filePosition <
getObjectMetadataResponse.ContentLength; i++)
    {
        CopyPartRequest copyPartRequest = new CopyPartRequest
        {
            UploadId = initResponse.UploadId,
            // Source.
            SourceBucket = existingBucketName,
            SourceKey = sourceKeyName,
            // Source object is stored using SSE-C. Provide encryption
information.
            CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
            CopySourceServerSideEncryptionCustomerProvidedKey =
base64Key, //"***source object encryption key ***",
            FirstByte = firstByte,
            // If the last part is smaller then our normal part size
then use the remaining size.
            LastByte = lastByte >
getObjectMetadataResponse.ContentLength ?
                getObjectMetadataResponse.ContentLength - 1 :
lastByte,

            // Target.
            DestinationBucket = existingBucketName,
            DestinationKey = targetKeyName,
            PartNumber = i,
            // Encryption information for the target object.
            ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
            ServerSideEncryptionCustomerProvidedKey = base64Key
        };
    }
```

```

        uploadResponses.Add(await
s3Client.CopyPartAsync(copyPartRequest));
        filePosition += partSize;
        firstByte += partSize;
        lastByte += partSize;
    }

    // Step 3: complete.
    CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
    {
        BucketName = existingBucketName,
        Key = targetKeyName,
        UploadId = initResponse.UploadId,
    };
    completeRequest.AddPartETags(uploadResponses);

    CompleteMultipartUploadResponse completeUploadResponse =
        await s3Client.CompleteMultipartUploadAsync(completeRequest);
}
catch (Exception exception)
{
    Console.WriteLine("Exception occurred: {0}", exception.Message);
    AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
    {
        BucketName = existingBucketName,
        Key = targetKeyName,
        UploadId = initResponse.UploadId
    };
    s3Client.AbortMultipartUpload(abortMPURequest);
}
}

private static async Task
CreateSampleObjUsingClientEncryptionKeyAsync(string base64Key, IAmazonS3
s3Client)
{
    // List to store upload part responses.
    List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

    // 1. Initialize.
    InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest

```

```
    {
        BucketName = existingBucketName,
        Key = sourceKeyName,
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key
    };

    InitiateMultipartUploadResponse initResponse =
        await s3Client.InitiateMultipartUploadAsync(initWithRequest);

    // 2. Upload Parts.
    long contentLength = new FileInfo(filePath).Length;
    long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

    try
    {
        long filePosition = 0;
        for (int i = 1; filePosition < contentLength; i++)
        {
            UploadPartRequest uploadRequest = new UploadPartRequest
            {
                BucketName = existingBucketName,
                Key = sourceKeyName,
                UploadId = initResponse.UploadId,
                PartNumber = i,
                PartSize = partSize,
                FilePosition = filePosition,
                FilePath = filePath,
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            // Upload part and add response to our list.
            uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));

            filePosition += partSize;
        }

        // Step 3: complete.
        CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
```

```
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
            UploadId = initResponse.UploadId,
            //PartETags = new List<PartETag>(uploadResponses)

        };
        completeRequest.AddPartETags(uploadResponses);

        CompleteMultipartUploadResponse completeUploadResponse =
            await s3Client.CompleteMultipartUploadAsync(completeRequest);
    }
    catch (Exception exception)
    {
        Console.WriteLine("Exception occurred: {0}", exception.Message);
        AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
            UploadId = initResponse.UploadId
        };
        await s3Client.AbortMultipartUploadAsync(abortMPURequest);
    }
}
}
```

Schützen von Daten mithilfe der clientseitigen Verschlüsselung

Clientseitige Verschlüsselung bezeichnet die lokale Verschlüsselung Ihrer Daten, um ihre Sicherheit bei der Übertragung und im Ruhezustand zu gewährleisten. Verwenden Sie den Amazon S3 Encryption Client, um Ihre Objekte zu verschlüsseln, bevor Sie sie an Amazon S3 senden. Wenn Ihre Objekte auf diese Weise verschlüsselt werden, werden Ihre Objekte keinen Dritten zugänglich gemacht, einschließlich AWS. Amazon S3 empfängt Ihre Objekte bereits verschlüsselt; Amazon S3 spielt beim Verschlüsseln oder Entschlüsseln Ihrer Objekte keine Rolle. Sie können sowohl den Amazon S3 Encryption Client als auch die [serverseitige Verschlüsselung](#) verwenden, um Ihre Daten zu verschlüsseln. Wenn Sie verschlüsselte Objekte an Amazon S3 senden, erkennt Amazon S3 die Objekte nicht als verschlüsselt, sondern nur typische Objekte.

Der Amazon S3 Encryption Client fungiert als Vermittler zwischen Ihnen und Amazon S3. Nachdem Sie den Amazon S3 Encryption Client instanziiert haben, werden Ihre Objekte als Teil Ihrer PutObject- und GetObject-Anforderungen in Amazon S3 automatisch verschlüsselt. Ihre Objekte werden alle mit einem eindeutigen Datenschlüssel verschlüsselt. Der Amazon S3 Encryption Client verwendet oder interagiert nicht mit Bucket-Schlüsseln, auch wenn Sie einen KMS Schlüssel als Ihren Wrapping-Schlüssel angeben.

Das Entwicklerhandbuch zum Amazon S3 Encryption Client konzentriert sich auf die Versionen 3.0 und höher des Amazon S3 Encryption Client. Weitere Informationen finden Sie unter [Was ist der Amazon S3 Encryption Client?](#) im Entwicklerhandbuch zum Amazon S3 Encryption Client.

Weitere Informationen zu früheren Versionen des Amazon S3 Encryption Clients finden Sie in AWS SDKEntwicklerhandbuch für Ihre Programmiersprache.

- [AWS SDK for Java](#)
- [AWS SDK for .NET](#)
- [AWS SDK for Go](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Ruby](#)
- [AWS SDK for C++](#)

Richtlinie für den Datenverkehr zwischen Netzwerken

In diesem Thema wird beschrieben, wie Amazon S3 Verbindungen vom Service zu anderen Speicherorten sichert.

Datenverkehr zwischen Service und lokalen Clients und Anwendungen

Die folgenden Verbindungen können kombiniert werden mit AWS PrivateLink um Konnektivität zwischen Ihrem privaten Netzwerk und AWS:

- Importieren in &S3; AWS Verbindung von Standort zu Standort. VPN Weitere Informationen finden Sie unter Was ist [AWS Site-to-Site VPN?](#)
- Importieren in &S3; AWS Direct Connect Verbindung. Weitere Informationen finden Sie unter [Was ist AWS Direct Connect?](#)

Der Zugriff auf Amazon S3 über das Netzwerk erfolgt über AWS veröffentlicht APIs. Clients müssen Transport Layer Security (TLS) 1.2 unterstützen. Wir empfehlen TLS 1.3. Kunden müssen auch Cipher Suites mit Perfect Forward Secrecy (PFS) unterstützen, wie Ephemeral Diffie-Hellman () oder Elliptic Curve Diffie-Hellman Ephemeral (DHE). ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi. Darüber hinaus müssen Sie Anfragen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signieren, die einem Prinzipal zugeordnet sind, oder Sie können den IAM [AWS Security Token Service \(STS\)](#), um temporäre Sicherheitsanmeldeinformationen zum Signieren von Anfragen zu generieren.

Verkehr zwischen AWS Ressourcen in derselben Region

Ein Virtual Private Cloud (VPC) -Endpunkt für Amazon S3 ist eine logische Einheit innerhalb einer VPC, die nur Konnektivität zu Amazon S3 ermöglicht. Der VPC leitet Anfragen an Amazon S3 weiter und leitet Antworten zurück an den VPC. Weitere Informationen finden Sie unter [VPC Endpoints](#) im VPC Benutzerhandbuch. Beispiele für Bucket-Richtlinien, mit denen Sie den Zugriff auf S3-Buckets von VPC Endpunkten aus steuern können, finden Sie unter [Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien](#)

AWS PrivateLink für Amazon S3

Mit AWS PrivateLink für Amazon S3 können Sie VPC Schnittstellenendpunkte (Schnittstellenendpunkte) in Ihrer virtuellen privaten Cloud (VPC) bereitstellen. Auf diese Endpunkte kann direkt von Anwendungen aus zugegriffen werden, die sich vor Ort befinden VPN AWS Direct Connect, oder in einem anderen AWS-Region VPC übertreiben.

Schnittstellenendpunkte werden durch eine oder mehrere elastische Netzwerkschnittstellen (ENIs) repräsentiert, denen private IP-Adressen aus Subnetzen in Ihrem zugewiesen werden. VPC Anforderungen an Amazon S3 über Schnittstellenendpunkte bleiben im Amazon-Netzwerk. Sie können auch über lokale Anwendungen auf Schnittstellen-Endpunkte in Ihren Anwendungen VPC zugreifen AWS Direct Connect or AWS Virtual Private Network (AWS VPN). Weitere Informationen darüber, wie Sie Ihr Netzwerk VPC mit Ihrem lokalen Netzwerk verbinden können, finden Sie in [AWS Direct Connect Benutzerhandbuch](#) und [AWS Site-to-Site VPN Benutzeranleitung](#).

Allgemeine Informationen zu Schnittstellenendpunkten finden Sie unter [VPC Schnittstellenendpunkte \(AWS PrivateLink\)](#) in der AWS PrivateLink Leitfaden.

Themen

- [Arten von VPC Endpunkten für Amazon S3](#)

- [Einschränkungen und Beschränkungen von AWS PrivateLink für Amazon S3](#)
- [Einen VPC Endpunkt erstellen](#)
- [Zugriff auf Amazon-S3-Schnittstellen-Endpunkte](#)
- [Privat DNS](#)
- [Zugriff auf Buckets, Access Points und Amazon S3 API Control-Operationen von Endpunkten mit S3-Schnittstelle](#)
- [Aktualisierung einer lokalen Konfiguration DNS](#)
- [Erstellen einer VPC Endpunktrichtlinie für Amazon S3](#)

Arten von VPC Endpunkten für Amazon S3

Sie können zwei Arten von VPC Endpunkten für den Zugriff auf Amazon S3 verwenden: Gateway-Endpunkte und Schnittstellen-Endpunkte (mit AWS PrivateLink). Ein Gateway-Endpunkt ist ein Gateway, das Sie in Ihrer Routing-Tabelle angeben, um auf Amazon S3 von Ihrem Computer VPC aus zuzugreifen. AWS Netzwerk. Schnittstellen-Endpunkte erweitern die Funktionalität von Gateway-Endpunkten, indem sie private IP-Adressen verwenden, um Anfragen an Amazon S3 von Ihrem Standort VPC, vor Ort oder von einem VPC anderen Ort aus weiterzuleiten AWS-Region mithilfe von Peering oder VPC AWS Transit Gateway. Weitere Informationen finden Sie unter [Was ist VPC Peering?](#) und [Transit Gateway im Vergleich zu VPC Peering](#).

Schnittstellenendpunkte sind mit Gateway-Endpunkten kompatibel. Wenn Sie bereits einen Gateway-Endpunkt in der haben VPC, können Sie beide Arten von Endpunkten gleichzeitig verwenden. VPC

Gateway-Endpunkte für Amazon S3	Schnittstellenendpunkte für Amazon S3
In beiden Fällen bleibt Ihr Netzwerkverkehr auf der AWS Netzwerk.	
Verwenden Sie öffentliche IP-Adressen von Amazon S3	Verwenden Sie private IP-Adressen von Ihrem VPC, um auf Amazon S3 zuzugreifen
Verwenden Sie dieselben Amazon S3 DNS S3-Namen	Endpunktspezifische Amazon S3 S3-Namen erforderlich DNS
Erlauben keinen On-Premises-Zugriff	Erlaubt On-Premises-Zugriff

Gateway-Endpunkte für Amazon S3	Schnittstellenendpunkte für Amazon S3
Erlauben Sie keinen Zugriff von einem anderen AWS-Region	Erlaube den Zugriff von einem VPC in einem anderen AWS-Region mithilfe von VPC Peering oder AWS Transit Gateway
Nicht berechnet	Berechnet

Weitere Informationen zu Gateway-Endpunkten finden Sie unter [VPCGateway-Endpunkte](#) im AWS PrivateLink Leitfaden.

Einschränkungen und Beschränkungen von AWS PrivateLink für Amazon S3

VPCEinschränkungen gelten für AWS PrivateLink für Amazon S3. Weitere Informationen finden Sie unter [Überlegungen zu Schnittstellenendpunkten](#) und [AWS PrivateLink Kontingente](#) in der AWS PrivateLink Leitfaden. Darüber hinaus gelten die folgenden Einschränkungen.

AWS PrivateLink für Amazon S3 unterstützt Folgendes nicht:

- [Endpunkte nach dem Federal Information Processing Standard \(FIPS\)](#)
- [Website-Endpunkte](#)
- [Globale Legacy-Endpunkte](#)
- [Endpunkte der S3-Dash-Region](#)
- [Amazon-S3-Dual-Stack-Endpunkte](#)
- Verwenden von [CopyObject](#) oder [UploadPartCopy](#) zwischen Buckets in verschiedenen AWS-Regionen
- Sicherheit auf Transportschicht (TLS) 1.1

Einen VPC Endpunkt erstellen

Informationen zum Erstellen eines VPC Schnittstellenendpunkts finden Sie unter [Erstellen eines VPC Endpunkts](#) in der AWS PrivateLink Leitfaden.

Zugriff auf Amazon-S3-Schnittstellen-Endpunkte

Wenn Sie einen Schnittstellenendpunkt erstellen, generiert Amazon S3 zwei Arten von endpunktspezifischen DNS S3-Namen: Regional und Zonal.

- Ein regionaler DNS Name umfasst eine eindeutige VPC Endpunkt-ID, eine Service-ID, AWS-Region, und `vpce.amazonaws.com` in seinem Namen. Für die VPC Endpunkt-ID `vpce-1a2b3c4d` könnte der generierte DNS Name beispielsweise ähnlich sein wie `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`.
- Ein zonaler DNS Name schließt die Availability Zone ein — zum Beispiel. `vpce-1a2b3c4d-5e6f-us-east-1a.s3.us-east-1.vpce.amazonaws.com` Sie können diese Option verwenden, wenn Ihre Architektur Availability Zones isoliert. Sie könnten sie beispielsweise zur Fehlereingrenzung oder zur Senkung der regionalen Datenübertragungskosten verwenden.

Endpunktspezifische DNS S3-Namen können von der öffentlichen S3-Domäne aus aufgelöst werden.
DNS

Privat DNS

Private DNS Optionen für VPC Schnittstellenendpunkte vereinfachen die Weiterleitung von S3-Verkehr über VPC Endpunkte und helfen Ihnen, den kostengünstigsten Netzwerkpfad zu nutzen, der für Ihre Anwendung verfügbar ist. Sie können private DNS Optionen verwenden, um regionalen S3-Verkehr weiterzuleiten, ohne Ihre S3-Clients so zu aktualisieren, dass sie die endpunktspezifischen DNS Namen Ihrer Schnittstellen-Endpunkte verwenden oder die Infrastruktur verwalten müssen. DNS Wenn private DNS Namen aktiviert sind, werden regionale DNS S3-Abfragen an die privaten IP-Adressen von AWS PrivateLink für die folgenden Endpunkte:

- Regionale Bucket-Endpunkte (z. B. `s3.us-east-1.amazonaws.com`)
- Kontrollendpunkte (z. B. `s3-control.us-east-1.amazonaws.com`)
- Zugriffspunkt-Endpunkte (z. B. `s3-accesspoint.us-east-1.amazonaws.com`)

AWS PrivateLink für Amazon S3 wird nicht unterstützt [Verwenden von Amazon-S3-Dual-Stack-Endpunkten](#). Wenn Sie einen DNS S3-Dual-Stack-Namen als privaten DNS Namen verwenden, wird Ihr IPv6 Datenverkehr entweder gelöscht oder, falls Ihre virtuelle private Cloud (VPC) über ein Internet-Gateway verfügt, wird Ihr IPv6 Datenverkehr über das Internet-Gateway in Ihrem weitergeleitet. VPC

Wenn Sie einen Gateway-Endpunkt in Ihrem VPC haben, können Sie eingehende VPC Anfragen automatisch über Ihren vorhandenen S3-Gateway-Endpunkt und lokale Anfragen über Ihren Schnittstellenendpunkt weiterleiten. Mit diesem Ansatz können Sie Ihre Nettwerkkosten optimieren, indem Sie Gateway-Endpunkte für Ihren eingehenden Datenverkehr verwenden, für die keine Gebühren anfallen. VPC Ihre lokalen Anwendungen können Folgendes verwenden AWS PrivateLink mit Hilfe des Resolver-Endpunkts für eingehende Anrufe. Amazon stellt Ihnen einen DNS Server namens Route 53 Resolver zur Verfügung VPC. Ein eingehender Resolver-Endpunkt leitet DNS Anfragen vom lokalen Netzwerk an den Route 53 Resolver weiter.

Important

Um bei der Verwendung von Enable Private DNS nur für eingehende Endpunkte den kostengünstigsten Netzwerkpfad nutzen zu können, muss in Ihrem System ein Gateway-Endpunkt vorhanden sein. VPC Das Vorhandensein eines Gateway-Endpunkts trägt dazu bei, dass ankommender VPC Datenverkehr immer über den AWS privaten Netzwerk, wenn die Option DNS Nur für eingehende Endpunkte privat aktivieren ausgewählt ist. Sie müssen diesen Gateway-Endpunkt beibehalten, solange Sie die Option DNS Nur für eingehende Endgeräte aktivieren aktiviert haben. Wenn Sie Ihren Gateway-Endpunkt löschen möchten, müssen Sie zunächst die Option Private Option DNS nur für eingehende Endpunkte aktivieren deaktivieren.

Wenn Sie einen vorhandenen Schnittstellenendpunkt auf Enable Private DNS only for Inbound Endpoints aktualisieren möchten, vergewissern Sie sich zunächst, dass Sie VPC über einen S3-Gateway-Endpunkt verfügen. Weitere Informationen zu Gateway-Endpunkten und zur Verwaltung privater DNS Namen finden Sie unter [VPC Gateway-Endpunkte bzw. Namen verwalten DNS](#) in der AWS PrivateLink Leitfaden.

Die Option DNS Nur privat für eingehende Endpunkte aktivieren ist nur für Dienste verfügbar, die Gateway-Endpunkte unterstützen.

Weitere Informationen zum Erstellen eines VPC Endpunkts, der Enable Private DNS nur für eingehende Endpunkte verwendet, finden Sie unter [Erstellen](#) eines Schnittstellenendpunkts in der AWS PrivateLink Leitfaden.

Die VPC Konsole verwenden

In der Konsole haben Sie zwei Optionen: DNS Name aktivieren und DNS Nur privat für eingehende Endpunkte aktivieren. DNS Name aktivieren ist eine Option, die unterstützt wird von AWS PrivateLink.

Mit der Option `DNSNames aktivieren` können Sie die private Konnektivität von Amazon zu Amazon S3 nutzen und gleichzeitig Anfragen an die standardmäßigen öffentlichen DNS Endpunktnamen stellen. Wenn diese Option aktiviert ist, können Kunden den für Ihre Anwendung kostengünstigsten Netzwerkpfad nutzen.

Wenn Sie private DNS Namen auf einem vorhandenen oder neuen VPC Schnittstellenendpunkt für Amazon S3 aktivieren, ist die Option `Private DNS nur für eingehende Endpunkte aktivieren` standardmäßig ausgewählt. Wenn diese Option ausgewählt ist, verwenden Ihre Anwendungen nur Schnittstellenendpunkte für Ihren On-Premises-Datenverkehr. Dieser eingehende VPC Datenverkehr verwendet automatisch die kostengünstigeren Gateway-Endpunkte. Alternativ können Sie die Option `Private DNS nur für eingehende Endpunkte aktivieren` deaktivieren, um alle S3-Anfragen über Ihren Schnittstellenendpunkt weiterzuleiten.

Verwenden Sie den AWS CLI

Wenn Sie keinen Wert für `PrivateDnsOnlyForInboundResolverEndpoint` angeben, wird standardmäßig `true` verwendet. Bevor Sie Ihre VPC Einstellungen übernehmen, wird jedoch überprüft, ob in der ein Gateway-Endpunkt vorhanden istVPC. Wenn in der ein Gateway-Endpunkt vorhanden istVPC, ist der Anruf erfolgreich. Wenn nicht, wird die folgende Fehlermeldung angezeigt:

Um den Wert `PrivateDnsOnlyForInboundResolverEndpoint` auf `true` zu setzen, VPC *vpce_id* muss über einen Gateway-Endpunkt für den Dienst verfügen.

Für einen neuen VPC Interface-Endpunkt

Verwenden Sie die `dns-options` Attribute `private-dns-enabled` und, um Private DNS über die Befehlszeile zu aktivieren. Die Option `PrivateDnsOnlyForInboundResolverEndpoint` im Attribut `dns-options` muss auf `true` gesetzt sein. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws ec2 create-vpc-endpoint \  
--region us-east-1 \  
--service-name s3-service-name \  
--vpc-id client-vpc-id \  
--subnet-ids client-subnet-id \  
--vpc-endpoint-type Interface \  
--private-dns-enabled \  
--ip-address-type ip-address-type \  
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true \  
--security-group-ids client-sg-id
```

Für einen vorhandenen VPC Endpunkt

Wenn Sie private DNS für einen vorhandenen VPC Endpunkt verwenden möchten, verwenden Sie den folgenden Beispielbefehl und ersetzen Sie den *user input placeholders* durch Ihre eigenen Informationen.

```
aws ec2 modify-vpc-endpoint \  
--region us-east-1 \  
--vpc-endpoint-id client-vpc-id \  
--private-dns-enabled \  
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=false
```

Wenn Sie einen vorhandenen VPC Endpunkt aktualisieren möchten, sodass Private DNS nur für den Inbound Resolver aktiviert wird, verwenden Sie das folgende Beispiel und ersetzen Sie die Beispielwerte durch Ihre eigenen.

```
aws ec2 modify-vpc-endpoint \  
--region us-east-1 \  
--vpc-endpoint-id client-vpc-id \  
--private-dns-enabled \  
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true
```

Zugriff auf Buckets, Access Points und Amazon S3 API Control-Operationen von Endpunkten mit S3-Schnittstelle

Sie können das AWS CLI or AWS SDKsum über S3-Schnittstellenendpunkte auf Buckets, S3-Zugriffspunkte und Amazon S3 API Control-Operationen zuzugreifen.

Die folgende Abbildung zeigt die Registerkarte „Details“ der VPC Konsole, auf der Sie den DNS Namen eines VPC Endpunkts finden. In diesem Beispiel lautet die VPCEndpunkt-ID (vpce-id) *vpce-0e25b8cdd720f900e* und der DNS Name lautet. **.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com*

Details	Subnets	Security Groups	Policy	Notifications	Tags
Endpoint ID	vpce-0e25b8cdd720f900e	VPC ID	vpce-0c0ccb9d87b1734bd VPCStack VPC	Status message	
Status	available	Service name	com.amazonaws.us-east-1.s3	DNS names	*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com (Z7HUB22UULQXV)
Creation time	January 8, 2021 at 1:30:11 AM UTC-8				
Endpoint type	Interface				

Wenn Sie den DNS Namen für den Zugriff auf eine Ressource verwenden, ersetzen Sie * mit dem entsprechenden Wert. Die entsprechenden Werte, die anstelle von * verwendet werden können, lauten wie folgt:

- bucket
- accesspoint
- control

Um beispielsweise auf einen Bucket zuzugreifen, verwenden Sie einen DNS Namen wie diesen:

```
bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com
```

Beispiele für die Verwendung von DNS Namen für den Zugriff auf Buckets, Access Points und Amazon S3 API Control-Operationen finden Sie in den folgenden Abschnitten von [AWS CLI Beispiele](#) und [AWS SDK Beispiele](#).

Weitere Informationen zum Anzeigen Ihrer endpunktspezifischen DNS Namen finden Sie unter [Konfiguration der privaten DNS Namen von Endpoint Services](#) anzeigen im VPC Benutzerhandbuch.

AWS CLI Beispiele

Für den Zugriff auf S3-Buckets, S3-Zugriffspunkte oder Amazon S3 API Control-Operationen über S3-Schnittstellenendpunkte in AWS CLI Befehle verwenden Sie die Parameter `--region` und `--endpoint-url`.

Beispiel: Verwenden Sie einen EndpunktURL, um Objekte in Ihrem Bucket aufzulisten

Ersetzen Sie im folgenden Beispiel den Bucket-Namen *my-bucketus-east-1*, Region und den DNS Namen der VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
aws s3 ls s3://my-bucket/ --region us-east-1 --endpoint-url  
https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```


Beispiel: Verwenden Sie einen EndpunktURL, um Objekte von einem Access Point aufzulisten

- Methode 1 — Verwendung des Amazon-Ressourcennamens (ARN) des Access Points mit dem Access Point-Endpunkt

Ersetzen Sie die ARN `us-east-1:123456789012:accesspoint/accesspointexamplename`, die Region `us-east-1` und die VPC Endpunkt-ID `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` durch Ihre eigenen Informationen.

```
aws s3api list-objects-v2 --bucket arn:aws:s3:us-east-1:123456789012:accesspoint/
accesspointexamplename --region us-east-1 --endpoint-url
https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Wenn Sie den Befehl nicht erfolgreich ausführen können, aktualisieren Sie AWS CLI auf die neueste Version und versuchen Sie es erneut. Weitere Informationen zu den Aktualisierungsanweisungen finden Sie unter [Installation oder Aktualisierung der neuesten Version von AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

- Methode 2 – Verwenden des Alias des Zugriffspunkts mit dem regionalen Bucket-Endpunkt

Ersetzen Sie im folgenden Beispiel den Access Point-Alias `accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias`, die Region `us-east-1` und die VPC Endpunkt-ID `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` durch Ihre eigenen Informationen.

```
aws s3api list-objects-v2 --
bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias
--region us-east-1 --endpoint-url https://bucket.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com
```

- Methode 3 – Verwenden des Alias des Zugriffspunkts mit dem Zugriffspunkt-Endpunkt

Um einen S3-Endpunkt zu erstellen, bei dem der Bucket als Teil des Hostnamens enthalten ist, legen Sie zunächst den Adressierungsstil auf `virtual` fest, damit `aws s3api` ihn verwenden kann. Weitere Informationen zu `AWS configure` finden Sie unter [Einstellungen für die Konfiguration und die Anmeldeinformationsdatei](#) im AWS Command Line Interface Benutzerleitfaden.

```
aws configure set default.s3.addressing_style virtual
```

Ersetzen Sie dann im folgenden Beispiel den Access Point-Alias *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen. Weitere Informationen zum Access Point-Alias finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt](#).

```
aws s3api list-objects-v2 --
bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias --
region us-east-1 --endpoint-url https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com
```

Beispiel: Verwenden Sie einen EndpunktURL, um Jobs mit einem API S3-Steuerungsvorgang aufzulisten

Ersetzen Sie im folgenden Beispiel die Region *us-east-1*, die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Konto-ID *12345678* durch Ihre eigenen Informationen.

```
aws s3control --region us-east-1 --endpoint-url
https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com list-jobs --
account-id 12345678
```

AWS SDKBeispiele

Für den Zugriff auf S3-Buckets, S3-Zugriffspunkte oder Amazon S3 API Control-Operationen über S3-Schnittstellenendpunkte bei Verwendung von AWS SDKs, aktualisieren Sie Ihre Version SDKs auf die neueste Version. Konfigurieren Sie dann Ihre Clients so, dass sie einen Endpunkt URL für den Zugriff auf einen Bucket, Access Point oder Amazon S3 API Control-Operationen über S3-Schnittstellenendpunkte verwenden.

SDK for Python (Boto3)

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Bucket zuzugreifen

Ersetzen Sie im folgenden Beispiel die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
s3_client = session.client(  
    service_name='s3',  
    region_name='us-east-1',  
    endpoint_url='https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'  
)
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Zugriffspunkt zuzugreifen

Ersetzen Sie im folgenden Beispiel die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
ap_client = session.client(  
    service_name='s3',  
    region_name='us-east-1',  
    endpoint_url='https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-  
east-1.vpce.amazonaws.com'  
)
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf Amazon S3 Control zuzugreifen API

Ersetzen Sie im folgenden Beispiel die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
control_client = session.client(  
    service_name='s3control',  
    region_name='us-east-1',  
    endpoint_url='https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'  
)
```

SDK for Java 1.x

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Bucket zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
// bucket client  
final AmazonS3 s3 = AmazonS3ClientBuilder.standard().withEndpointConfiguration(  

```

```
        new AwsClientBuilder.EndpointConfiguration(
            "https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",
            Regions.DEFAULT_REGION.getName()
        )
    ).build();
List<Bucket> buckets = s3.listBuckets();
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Zugriffspunkt zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und ARN *us-east-1:123456789012:accesspoint/prod* durch Ihre eigenen Informationen.

```
// accesspoint client
final AmazonS3 s3accesspoint =
    AmazonS3ClientBuilder.standard().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
            "https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",
            Regions.DEFAULT_REGION.getName()
        )
    ).build();
ObjectListing objects = s3accesspoint.listObjects("arn:aws:s3:us-east-1:123456789012:accesspoint/prod");
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen Amazon S3 API Control-Vorgang zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
// control client
final AWSS3Control s3control =
    AWSS3ControlClient.builder().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
            "https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",
            Regions.DEFAULT_REGION.getName()
        )
    ).build();
final ListJobsResult jobs = s3control.listJobs(new ListJobsRequest());
```

SDK for Java 2.x

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Bucket zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Region *Region.US_EAST_1* durch Ihre eigenen Informationen.

```
// bucket client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)

    .endpointOverride(URI.create("https://bucket.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
    .build()
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Zugriffspunkt zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Region *Region.US_EAST_1* durch Ihre eigenen Informationen.

```
// accesspoint client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)

    .endpointOverride(URI.create("https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
    .build()
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf Amazon S3 Control zuzugreifen API

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Region *Region.US_EAST_1* durch Ihre eigenen Informationen.

```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

    .endpointOverride(URI.create("https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
```

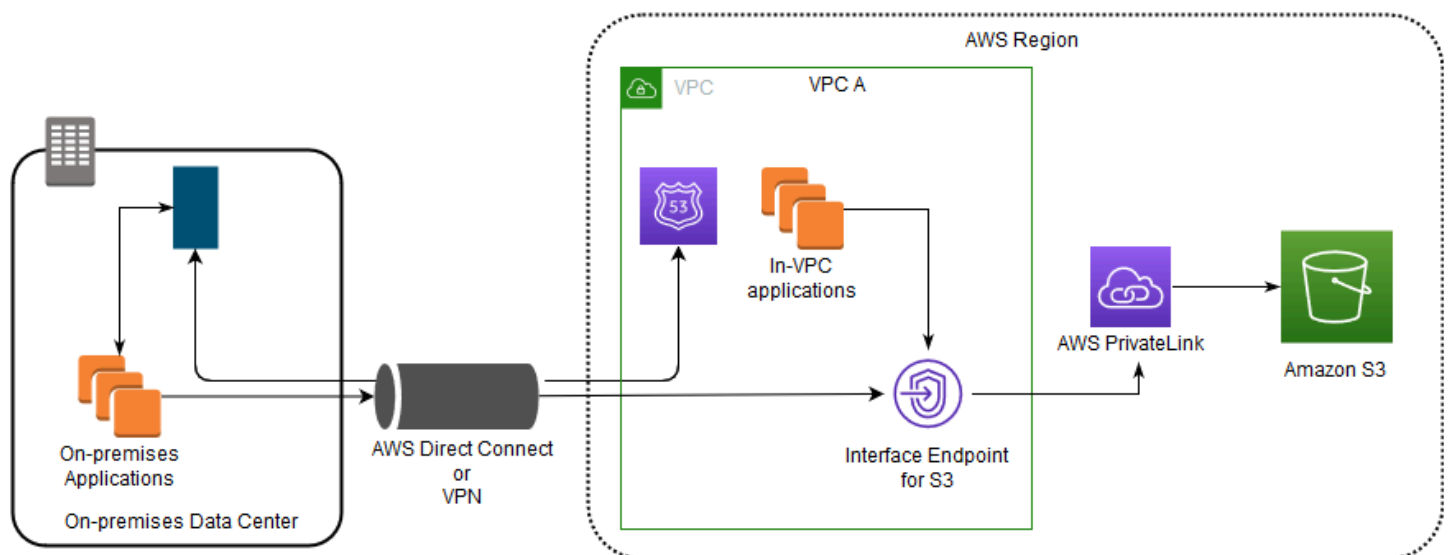
.build()

Aktualisierung einer lokalen Konfiguration DNS

Wenn Sie endpunktspezifische DNS Namen für den Zugriff auf die Schnittstellenendpunkte für Amazon S3 verwenden, müssen Sie Ihren lokalen Resolver nicht aktualisieren. DNS Sie können den endpunktspezifischen DNS Namen mit der privaten IP-Adresse des Schnittstellenendpunkts aus der öffentlichen Amazon S3 DNS S3-Domain auflösen.

Verwenden von Schnittstellenendpunkten für den Zugriff auf Amazon S3 ohne einen Gateway-Endpunkt oder ein Internet-Gateway in der VPC

Schnittstellenendpunkte in Ihrem VPC System können sowohl interne als auch lokale VPC Anwendungen über das Amazon-Netzwerk an Amazon S3 weiterleiten, wie in der folgenden Abbildung dargestellt.



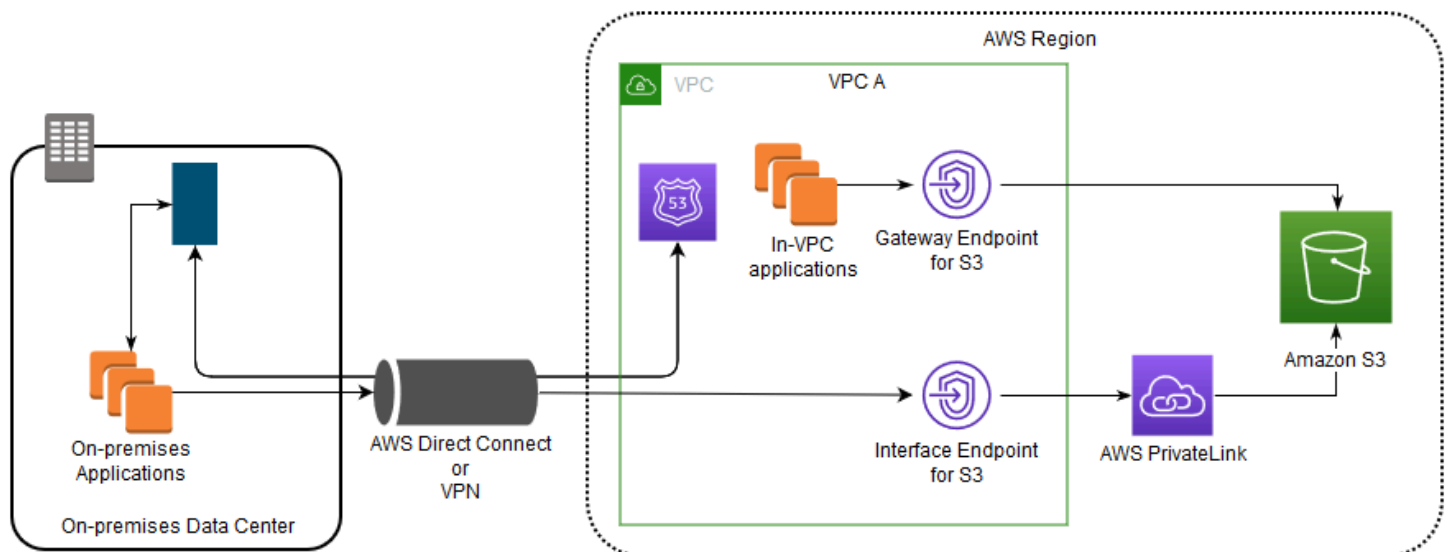
Das Diagramm veranschaulicht folgende Vorgänge:

- Ihr lokales Netzwerk verwendet AWS Direct Connect oder AWS VPN um eine Verbindung zu A herzustellen. VPC
- Ihre Anwendungen vor Ort und in VPC A verwenden endpunktspezifische DNS Namen, um über den Endpunkt der S3-Schnittstelle auf Amazon S3 zuzugreifen.
- Lokale Anwendungen senden dabei Daten an den Schnittstellenendpunkt VPC AWS Direct Connect (oder AWS VPN). AWS PrivateLink verschiebt die Daten vom Schnittstellenendpunkt zu Amazon S3 über den AWS Netzwerk.

- VPCIn-Anwendungen senden auch Datenverkehr an den Schnittstellenendpunkt. AWS PrivateLink verschiebt die Daten vom Schnittstellenendpunkt zu Amazon S3 über den AWS Netzwerk.

Gemeinsame Verwendung von Gateway-Endpunkten und Schnittstellen-Endpunkten für den VPC Zugriff auf Amazon S3

Sie können Schnittstellen-Endpunkte erstellen und den vorhandenen Gateway-Endpunkt darin beibehalten VPC, wie das folgende Diagramm zeigt. Mit diesem Ansatz ermöglichen Sie internen VPC Anwendungen, weiterhin über den Gateway-Endpunkt auf Amazon S3 zuzugreifen, was nicht in Rechnung gestellt wird. Dann würden nur Ihre lokalen Anwendungen Schnittstellenendpunkte für den Zugriff auf Amazon S3 verwenden. Um auf diese Weise auf Amazon S3 zuzugreifen, müssen Sie Ihre lokalen Anwendungen so aktualisieren, dass sie endpunktspezifische DNS Namen für Amazon S3 verwenden.



Das Diagramm veranschaulicht folgende Vorgänge:

- Lokale Anwendungen verwenden endpunktspezifische DNS Namen, um Daten innerhalb des Durchgangs an den Schnittstellenendpunkt zu senden VPC AWS Direct Connect (oder AWS VPN). AWS PrivateLink verschiebt die Daten vom Schnittstellenendpunkt zu Amazon S3 über den AWS Netzwerk.
- Mithilfe der standardmäßigen regionalen Amazon S3-Namen senden VPC In-Anwendungen Daten an den Gateway-Endpunkt, der eine Verbindung zu Amazon S3 herstellt über AWS Netzwerk.

Weitere Informationen zu Gateway-Endpunkten finden Sie unter [VPCGateway-Endpunkte](#) im VPCBenutzerhandbuch.

Erstellen einer VPC Endpunktrichtlinie für Amazon S3

Sie können Ihrem VPC Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf Amazon S3 steuert. Die Richtlinie gibt die folgenden Informationen an:

- Das Tool AWS Identity and Access Management (IAM) Principal, der Aktionen ausführen kann
- Aktionen, die ausgeführt werden können
- Ressourcen, für die Aktionen ausgeführt werden können

Sie können auch Amazon S3 S3-Bucket-Richtlinien verwenden, um den Zugriff auf bestimmte Buckets von einem bestimmten VPC Endpunkt aus zu beschränken, indem Sie die `aws:sourceVpce` Bedingung in Ihrer Bucket-Richtlinie verwenden. Die folgenden Beispiele zeigen Richtlinien, die den Zugriff auf einen Bucket oder einen Endpunkt einschränken.

Themen

- [Beispiel: Beschränken des Zugriffs auf einen bestimmten Bucket von einem Endpunkt aus VPC](#)
- [Beispiel: Beschränken des Zugriffs auf Buckets in einem bestimmten Konto von einem Endpunkt aus VPC](#)
- [Beispiel: Beschränkung des Zugriffs auf einen bestimmten VPC Endpunkt in der S3-Bucket-Richtlinie](#)

Beispiel: Beschränken des Zugriffs auf einen bestimmten Bucket von einem Endpunkt aus VPC

Sie können eine Endpunktrichtlinie erstellen, die den Zugriff auf spezifische Amazon-S3-Buckets beschränkt. Diese Art von Richtlinie ist nützlich, wenn Sie andere AWS-Services in deinem VPC, das Eimer benutzt. Die folgende Bucket-Richtlinie schränkt den Zugriff auf ausschließlich `amzn-s3-demo-bucket1` ein. Um diese Endpunktrichtlinie zu verwenden, ersetzen Sie `amzn-s3-demo-bucket1` im Beispiel durch den Namen Ihres Buckets.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    { "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
```



```

    "s3:GetObject",
    "s3:PutObject"
  ],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket1",
               "arn:aws:s3:::amzn-s3-demo-bucket1/*"]
}
]
}

```

Beispiel: Beschränken des Zugriffs auf Buckets in einem bestimmten Konto von einem Endpunkt aus VPC

Sie können eine Endpunktrichtlinie erstellen, die den Zugriff nur auf die S3-Buckets in einem bestimmten Bereich beschränkt AWS-Konto. Um zu verhindern, dass Ihre VPC Clients auf Buckets zugreifen, die Sie nicht besitzen, verwenden Sie die folgende Anweisung in Ihrer Endpunktrichtlinie. Die folgende Beispielanweisung erstellt eine Richtlinie, die den Zugriff auf Ressourcen einschränkt, die einer einzelnen Person gehören AWS-Konto ID, **111122223333**.

```

{
  "Statement": [
    {
      "Sid": "Access-to-bucket-in-specific-account-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}

```


 Note

Um die zu spezifizieren AWS-Konto ID der Ressource, auf die zugegriffen wird. Sie können entweder den `aws:ResourceAccount` oder den `s3:ResourceAccount` Schlüssel in Ihrer IAM Richtlinie verwenden. Beachten Sie jedoch, dass einige AWS-Services verlassen sich auf den Zugang zu AWS verwaltete Buckets. Daher kann sich die Verwendung der `s3:ResourceAccount` Taste `aws:ResourceAccount` oder in Ihrer IAM Richtlinie auch auf den Zugriff auf diese Ressourcen auswirken.

Beispiel: Beschränkung des Zugriffs auf einen bestimmten VPC Endpunkt in der S3-Bucket-Richtlinie

Beispiel: Beschränkung des Zugriffs auf einen bestimmten VPC Endpunkt in der S3-Bucket-Richtlinie

Die folgende Amazon S3 S3-Bucket-Richtlinie ermöglicht den Zugriff auf einen bestimmten Bucket nur vom VPC Endpunkt aus `vpce-1a2b3c4d`. `amzn-s3-demo-bucket2` Die Richtlinie lehnt sämtlichen Zugriff auf den Bucket ab, der nicht über den angegebenen Endpunkt erfolgt. Die `aws:sourceVpce` Bedingung spezifiziert den Endpunkt und erfordert keinen Amazon-Ressourcennamen (ARN) für die VPC Endpunktressource, sondern nur die Endpunkt-ID. Um diese Bucket-Richtlinie zu verwenden, ersetzen Sie `amzn-s3-demo-bucket2` und `vpce-1a2b3c4d` durch den Namen Ihres Buckets und Ihren Endpunkt.

 Important

- Wenn Sie die folgende Amazon S3 S3-Bucket-Richtlinie anwenden, um den Zugriff nur auf bestimmte VPC Endpunkte zu beschränken, blockieren Sie möglicherweise Ihren Zugriff auf den Bucket, ohne dies zu beabsichtigen. Bucket-Richtlinien, mit denen der Bucket-Zugriff speziell auf Verbindungen beschränkt werden soll, die von Ihrem VPC Endpunkt ausgehen, können alle Verbindungen zum Bucket blockieren. Informationen zur Behebung dieses Problems finden Sie unter [Meine Bucket-Richtlinie hat die falsche VPC oder die VPC Endpunkt-ID. Wie kann ich die Richtlinie korrigieren, sodass ich auf den Bucket zugreifen kann?](#) in der AWS Support Wissenszentrum.
- Bevor Sie die folgende Beispielrichtlinie verwenden, ersetzen Sie die VPC Endpunkt-ID durch einen Wert, der Ihrem Anwendungsfall entspricht. Andernfalls können Sie nicht auf Ihren Bucket zugreifen.

- Diese Richtlinie deaktiviert den Konsolenzugriff auf den angegebenen Bucket, da Konsolenanfragen nicht vom angegebenen VPC Endpunkt stammen.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    { "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket2",
                  "arn:aws:s3:::amzn-s3-demo-bucket2/*"],
      "Condition": {"StringNotEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}}
    }
  ]
}
```

Weitere Richtlinienbeispiele finden Sie unter [Endpoints for Amazon S3](#) im VPCBenutzerhandbuch.

Weitere Informationen zur VPC Konnektivität finden Sie unter [VPCNetzwerk-zu-Konnektivitätsoptionen](#) im AWS Whitepaper [Amazon Virtual Private Cloud Cloud-Konnektivitätsoptionen](#).

Zugriffsverwaltung

In AWS, eine Ressource ist eine Entität, mit der Sie arbeiten können. In Amazon Simple Storage Service (S3) sind Buckets und Objekte die ursprünglichen Amazon S3 S3-Ressourcen. Jeder S3-Kunde hat wahrscheinlich Buckets mit Objekten darin. Als S3 um neue Funktionen erweitert wurde, kamen auch zusätzliche Ressourcen hinzu, aber nicht jeder Kunde nutzt diese funktionspezifischen Ressourcen. Weitere Informationen zu Amazon S3 S3-Ressourcen finden Sie unter [S3-Ressourcen](#).

Standardmäßig werden alle Amazon S3-Ressourcen als privat eingestuft. Standardmäßig ist der Root-Benutzer des AWS-Konto der die Ressource erstellt hat (Ressourcenbesitzer), und IAM Benutzer innerhalb dieses Kontos mit den erforderlichen Berechtigungen können auf eine von ihnen erstellte Ressource zugreifen. Der Besitzer der Ressource entscheidet, wer sonst noch auf die Ressource zugreifen kann und welche Aktionen andere an der Ressource ausführen dürfen. S3

verfügt über verschiedene Tools zur Zugriffsverwaltung, mit denen Sie anderen Zugriff auf Ihre S3-Ressourcen gewähren können.

Die folgenden Abschnitte bieten Ihnen einen Überblick über S3-Ressourcen, die verfügbaren S3-Zugriffsverwaltungstools und die besten Anwendungsfälle für die einzelnen Zugriffsverwaltungstools. Die Listen in diesen Abschnitten sollen umfassend sein und alle S3-Ressourcen, Zugriffsverwaltungstools und allgemeine Anwendungsfälle für die Zugriffsverwaltung enthalten. Gleichzeitig sind diese Abschnitte als Verzeichnisse konzipiert, die Sie zu den gewünschten technischen Details führen. Wenn Sie sich mit einigen der folgenden Themen gut auskennen, können Sie zu dem Abschnitt springen, der für Sie relevant ist.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Topics

- [S3-Ressourcen](#)
- [Identitäten](#)
- [Tools zur Zugriffsverwaltung](#)
- [Aktionen](#)
- [Anwendungsfälle für die Zugriffsverwaltung](#)
- [Fehlerbehebung bei der Zugriffsverwaltung](#)

S3-Ressourcen

Die ursprünglichen Amazon S3 S3-Ressourcen sind Buckets und die Objekte, die sie enthalten. Wenn neue Funktionen zu S3 hinzugefügt werden, kommen auch neue Ressourcen hinzu. Im Folgenden finden Sie eine vollständige Liste der S3-Ressourcen und ihrer jeweiligen Funktionen.

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
bucket	Kern-Features	Ein Bucket ist ein Container für Objekte. Um ein Objekt in S3 zu speichern, erstellen Sie einen Bucket und laden Sie dann ein oder mehrere Objekte in den Bucket hoch. Weitere Informationen finden Sie unter Erstellen, Konfigurieren und Arbeiten mit Amazon S3-Buckets .

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
object		<p>Ein Objekt kann eine Datei und alle Metadaten sein, die diese Datei beschreiben. Wenn sich ein Objekt im Bucket befindet, können Sie es öffnen, herunterladen und verschieben. Weitere Informationen finden Sie unter Hochladen, Herunterladen und Arbeiten mit Objekten in Amazon S3.</p>
accesspoint	Zugriffspunkte	<p>Access Points sind benannte Netzwerkendpunkte, die an Buckets angehängt sind, mit denen Sie Amazon S3 S3-Objektoperationen wie <code>GetObject</code> und <code>PutObject</code> ausführen können. Jeder Access Point verfügt über eigene Berechtigungen, Netzwerksteuerungen und eine benutzerdefinierte Zugriffspunktrichtlinie, die mit der Bucket-Richtlinie zusammenarbeitet, die dem zugrunde liegenden Bucket zugeordnet ist. Sie können jeden Access Point so konfigurieren, dass er nur Anfragen von einer Virtual Private Cloud (VPC) akzeptiert, oder benutzerdefinierte Einstellungen für den Block Public Access für jeden Access Point konfigurieren. Weitere Informationen finden Sie unter Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten.</p>
objectlambdaaccesspoint		<p>Ein Object Lambda Access Point ist ein Access Point für einen Bucket, der auch mit einer Lambda-Funktion verknüpft ist. Mit Object Lambda Access Point können Sie Amazon S3 Ihren eigenen Code und HEAD Anfragen zur Änderung und Verarbeitung von Daten hinzufügen <code>GETLIST</code>, wenn diese an eine Anwendung zurückgegeben werden. Weitere Informationen finden Sie unter Erstellen von Objekt-Lambda-Zugriffspunkten.</p>

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
multiregionaccesspoint		<p>Multi-Region-Access Points bieten einen globalen Endpunkt, über den Anwendungen Anfragen von Amazon S3 S3-Buckets bearbeiten können, die sich in mehreren AWS Regionen. Sie können Multi-Regions Access Points verwenden, um Multi-Regions-Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Anstatt Anfragen über das überlastete öffentliche Internet zu senden, werden Anwendungsanfragen an einen globalen Access Point-Endpunkt mit mehreren Regionen automatisch über den AWS globales Netzwerk zum nächstgelegenen Amazon S3 S3-Bucket. Weitere Informationen finden Sie unter Multi-Regions-Zugriffspunkte in Amazon S3.</p>
job	S3-BatchVorgänge	<p>Ein Job ist eine Ressource der S3-Funktion Batch Operations. Sie können S3 Batch Operations verwenden, um umfangreiche Batch-Operationen für Listen von Amazon S3 S3-Objekten durchzuführen, die Sie angeben. Amazon S3 verfolgt den Fortschritt des Batch-Vorgangs, sendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht über alle Aktionen, sodass Sie ein vollständig verwaltetes, überprüfbares und serverloses Erlebnis erhalten. Weitere Informationen finden Sie unter Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch..</p>

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
storagele nsconfigu ration	S3 Storage Lens	Eine S3 Storage Lens-Konfiguration erfasst unternehm ensweite Speichermetriken und Benutzerdaten für alle Konten. S3 Storage Lens bietet Administratoren eine zentrale Ansicht der Objektspeichernutzung und der Aktivität en von Hunderten oder sogar Tausenden von Konten in einem Unternehmen mit Details, um Erkenntnisse auf mehreren Aggregationsebenen zu gewinnen. Weitere Informationen finden Sie unter Bewerten Ihrer Speichera ktivität und -nutzung mit Amazon S3 Storage Lens .
storagele nsgroup		Eine S3 Storage Lens-Gruppe aggregiert Metriken mithilfe benutzerdefinierter Filter, die auf Objektmetadaten basieren. Mithilfe von S3 Storage Lens-Gruppen können Sie Merkmale Ihrer Daten untersuchen, z. B. die Verteilun g von Objekten nach Alter, Ihre häufigsten Dateitypen und mehr. Weitere Informationen finden Sie unter Arbeiten mit S3-Storage-Lens-Gruppen .
accessgra ntsinstan ce	S3 Access Grants	Eine S3 Access Grants-Instanz ist ein Container für die von Ihnen erstellten S3-Grants. Mit S3 Access Grants können Sie Berechtigungen für Ihre Amazon S3 S3-Daten für IAM Identitäten in Ihrem Konto, IAM Identitäten in anderen Konten (kontoübergreifend) und Verzeichnisidentitäten, die hinzugefügt wurden, erstellen AWS IAM Identity Center aus Ihrem Unternehmensverzeichnis. Weitere Informationen zu S3 Access Grants finden Sie unter Verwalten des Zugriffs mit S3-Zugriffsberechtigungen .

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
accessgrantslocation		<p>Ein Access Grants-Standort ist ein Bucket, ein Präfix innerhalb eines Buckets oder ein Objekt, das Sie in Ihrer S3 Access Grants-Instanz registrieren. Sie müssen Standorte innerhalb der S3 Access Grants-Instanz registrieren, bevor Sie einen Zuschuss für diesen Standort erstellen können. Anschließend können Sie mit S3 Access Grants Zugriff auf den Bucket, das Präfix oder das Objekt für IAM Identitäten in Ihrem Konto, IAM Identitäten in anderen Konten (kontoübergreifend) und Verzeichnisidentitäten, die hinzugefügt wurden, gewähren AWS IAM Identity Center aus Ihrem Unternehmensverzeichnis. Weitere Informationen zu S3 Access Grants finden Sie unter Verwalten des Zugriffs mit S3-Zugriffsberechtigungen</p>
accessgrant		<p>Ein Access Grant ist eine individuelle Gewährung Ihrer Amazon S3 S3-Daten. Mit S3 Access Grants können Sie Berechtigungen für Ihre Amazon S3 S3-Daten für IAM Identitäten in Ihrem Konto, IAM Identitäten in anderen Konten (kontoübergreifend) und Verzeichnisidentitäten, die hinzugefügt wurden, erstellen AWS IAM Identity Center aus Ihrem Unternehmensverzeichnis. Weitere Informationen zu S3 Access Grants finden Sie unter Verwalten des Zugriffs mit S3-Zugriffsberechtigungen</p>

Buckets

Es gibt zwei Arten von Amazon S3 S3-Buckets: Allzweck-Buckets und Directory-Buckets.

- Allzweck-Buckets sind der ursprüngliche S3-Bucket-Typ und werden für die meisten Anwendungsfälle und Zugriffsmuster empfohlen. Allzweck-Buckets ermöglichen auch Objekte, die in allen Speicherklassen gespeichert werden, mit Ausnahme von S3 Express One Zone. Weitere Informationen zu S3-Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#)

- Directory-Buckets verwenden die Speicherklasse S3 Express One Zone. Dies wird empfohlen, wenn Ihre Anwendung leistungsempfindlich ist und von einstelligen Millisekunden und Latenzen profitiert. PUT GET Weitere Informationen finden Sie unter [Verzeichnis-Buckets](#), [Was ist S3 Express One Zone?](#) und [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

Kategorisieren von S3-Ressourcen

Amazon S3 bietet Funktionen zur Kategorisierung und Organisation Ihrer S3-Ressourcen. Die Kategorisierung Ihrer Ressourcen ist nicht nur nützlich, um sie zu organisieren, sondern Sie können auch Zugriffsverwaltungsregeln auf der Grundlage der Ressourcenkategorien festlegen. Insbesondere Präfixe und Tagging sind zwei Funktionen zur Speicherorganisation, die Sie bei der Festlegung von Zugriffsverwaltungsberechtigungen verwenden können.

Note

Die folgenden Informationen beziehen sich auf Allzweck-Buckets. Verzeichnis-Buckets unterstützen kein Tagging und sie haben Präfixbeschränkungen. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

- Präfixe — Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, die verwendet wird, um die Objekte zu organisieren, die in Ihren S3-Buckets gespeichert sind. Sie können ein Trennzeichen verwenden, z. B. einen Schrägstrich (/), um das Ende des Präfixes innerhalb des Objektschlüsselnamens anzugeben. Beispielsweise können Sie Objektschlüsselnamen verwenden, die mit dem `engineering/` Präfix beginnen, oder Objektschlüsselnamen, die mit dem Präfix `marketing/campaigns/` beginnen. Die Verwendung eines Trenzeichens am Ende Ihres Präfixes, z. B. eines Schrägstrichs, / emuliert die Benennungskonventionen für Ordner und Dateien. In S3 ist das Präfix jedoch Teil des Objektschlüsselnamens. In S3-Buckets für allgemeine Zwecke gibt es keine tatsächliche Ordnerhierarchie.

Amazon S3 unterstützt das Organisieren und Gruppieren von Objekten mithilfe ihrer Präfixe. Sie können den Zugriff auf Objekte auch anhand ihrer Präfixe verwalten. Sie können beispielsweise den Zugriff auf Objekte beschränken, deren Namen mit einem bestimmten Präfix beginnen.

Weitere Informationen finden Sie unter [Organisieren von Objekten mit Präfixen](#). Die S3-Konsole verwendet das Konzept von Ordnern, bei denen es sich bei Buckets für allgemeine Zwecke im Wesentlichen um Präfixe handelt, die dem Objektschlüsselnamen vorangestellt werden. Weitere

Informationen finden Sie unter [Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern](#).

- Tags — Jedes Tag ist ein Schlüssel-Wert-Paar, das Sie Ressourcen zuweisen. Sie können beispielsweise einige Ressourcen mit dem Tag kennzeichnen. `topicCategory=engineering` Sie können Tagging verwenden, um die Kostenzuweisung, Kategorisierung und Organisation sowie die Zugriffskontrolle zu erleichtern. Bucket-Tagging wird nur für die Kostenzuweisung verwendet. Sie können Objekte, S3 Storage Lens, Jobs und S3 Access Grants zu Organisationszwecken oder zur Zugriffskontrolle taggen. In S3 Access Grants können Sie Tagging auch für die Kostenzuweisung verwenden. Als Beispiel für die Steuerung des Zugriffs auf Ressourcen mithilfe ihrer Tags können Sie nur Objekte teilen, die über ein bestimmtes Tag oder eine Kombination von Tags verfügen.

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf AWS Ressourcen mithilfe von Ressourcen-Tags](#) im IAMBenutzerhandbuch.

Identitäten

In Amazon S3 ist der Ressourcenbesitzer die Identität, die die Ressource erstellt hat, z. B. ein Bucket oder ein Objekt. Standardmäßig können nur der Root-Benutzer des Kontos, das die Ressource erstellt hat, und IAM Identitäten innerhalb des Kontos, die über die erforderlichen Berechtigungen verfügen, auf die S3-Ressource zugreifen. Ressourcenbesitzer können anderen Identitäten Zugriff auf ihre S3-Ressourcen gewähren.

Identitäten, denen eine Ressource nicht gehört, können Zugriff auf diese Ressource beantragen. Anfragen an eine Ressource sind entweder authentifiziert oder nicht authentifiziert. Authentifizierte Anfragen müssen einen Signaturwert enthalten, der den Absender der Anfrage authentifiziert. Für nicht authentifizierte Anfragen ist jedoch keine Signatur erforderlich. Wir empfehlen, dass Sie nur authentifizierten Benutzern Zugriff gewähren. Weitere Informationen über Anforderungsauthentifizierung finden Sie unter [Senden von Anforderungen](#).

Important

Wir empfehlen Ihnen, die nicht zu verwenden AWS-Konto Root-Benutzeranmeldedaten, um authentifizierte Anfragen zu stellen. Erstellen Sie stattdessen eine IAM Rolle und gewähren Sie dieser Rolle vollen Zugriff. Wir bezeichnen Benutzer mit dieser Rolle als Administratorbenutzer. Sie können anstelle von Anmeldeinformationen, die der Administratorrolle zugewiesen sind, verwenden AWS-Konto Root-Benutzeranmeldedaten, mit

denen Sie interagieren AWS und Aufgaben wie das Erstellen eines Buckets, das Erstellen von Benutzern und das Erteilen von Berechtigungen ausführen. Weitere Informationen finden Sie unter [AWS-Konto Root-Benutzeranmeldedaten und IAM Benutzeranmeldedaten](#) in Allgemeine AWS-Referenz, und finden Sie [bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch](#).

Es kann sich um eine der folgenden Identitäten handeln, die auf Ihre Daten in Amazon S3 zugreifen:

AWS-Konto owner

Das Tool AWS-Konto das die Ressource erstellt hat. Zum Beispiel das Konto, das den Bucket erstellt hat. Dieses Konto besitzt die Ressource. Weitere Informationen finden Sie unter [AWS Konto Root-Benutzer](#).

IAM Identitäten im selben Konto des AWS-Konto owner

Bei der Einrichtung von Konten für neue Teammitglieder, die S3-Zugriff benötigen, AWS-Konto Der Besitzer kann verwenden AWS Identity and Access Management (IAM), um [Benutzer](#), [Gruppen](#) und [Rollen](#) zu erstellen. Das Tool AWS-Konto Der Besitzer kann dann Ressourcen mit diesen IAM Identitäten teilen. Der Kontoinhaber kann auch die Berechtigungen angeben, mit denen die IAM Identitäten vergeben werden sollen, wodurch die Aktionen, die mit den gemeinsam genutzten Ressourcen ausgeführt werden können, zugelassen oder verweigert werden.

IAM Identitäten bieten erweiterte Funktionen, einschließlich der Möglichkeit, von Benutzern die Eingabe von Anmeldeinformationen zu verlangen, bevor sie auf gemeinsam genutzte Ressourcen zugreifen können. Durch die Verwendung von IAM Identitäten können Sie eine Form der IAM Multi-Faktor-Authentifizierung (MFA) implementieren, um eine starke Identitätsbasis zu unterstützen. Eine IAM bewährte Methode besteht darin, Rollen für die Zugriffsverwaltung zu erstellen, anstatt jedem einzelnen Benutzer Berechtigungen zu erteilen. Sie weisen einzelnen Benutzern die entsprechende Rolle zu. Weitere Informationen finden Sie unter [Bewährte Sicherheitsmethoden unter IAM](#).

Sonstige AWS Kontoinhaber und ihre IAM Identitäten (kontoübergreifender Zugriff)

Das Tool AWS-Konto Der Besitzer kann auch andere angeben AWS Kontoinhaber oder IAM Identitäten, die einem anderen gehören AWS Konto, Zugriff auf Ressourcen.

Note

Delegierung von Genehmigungen — Wenn ein AWS-Konto besitzt eine Ressource, kann sie diese Berechtigungen einer anderen Person gewähren AWS-Konto. Dieses Konto kann dann diese Berechtigungen oder einen Teil davon an Benutzer desselben Kontos delegieren. Dies wird auch als Berechtigungsdelegation bezeichnet. Ein Konto, das Berechtigungen von einem anderen Konto erhält, kann diese Berechtigungen jedoch nicht „kontoübergreifend“ an ein anderes delegieren AWS-Konto.

Anonyme Benutzer (öffentlicher Zugriff)

Das Tool AWS-Konto Der Besitzer kann Ressourcen veröffentlichen. Wenn eine Ressource veröffentlicht wird, wird die Ressource technisch gesehen mit dem anonymen Benutzer geteilt. Buckets, die seit April 2023 erstellt wurden, blockieren standardmäßig den gesamten öffentlichen Zugriff, sofern Sie diese Einstellung nicht ändern. Wir empfehlen, dass Sie Ihre Buckets so einrichten, dass sie den öffentlichen Zugriff blockieren, und dass Sie nur authentifizierten Benutzern Zugriff gewähren. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

AWS-Services

Der Besitzer der Ressource kann einen anderen gewähren AWS Servicezugriff auf eine Amazon S3 S3-Ressource. Sie können zum Beispiel den gewähren AWS CloudTrail s3:PutObjectServiceberechtigung zum Schreiben von Protokolldateien in Ihren Bucket. Weitere Informationen finden Sie unter Gewähren [des Zugriffs auf ein AWS Dienst](#).

Identitäten im Unternehmensverzeichnis

Der Ressourcenbesitzer kann Benutzern oder Rollen aus Ihrem Unternehmensverzeichnis mithilfe von S3 Access Grants Zugriff auf eine [S3-Ressource gewähren](#). Weitere Informationen zum Hinzufügen Ihres Unternehmensverzeichnisses zu AWS IAM Identity Center, siehe [Was ist IAM Identity Center?](#) .

Besitzer von Buckets oder Ressourcen

Das Tool AWS-Konto die Sie verwenden, um Buckets zu erstellen und Objekte hochzuladen, besitzt diese Ressourcen. Ein Bucket-Besitzer kann einem anderen Benutzer kontoübergreifende Berechtigungen gewähren AWS-Konto (oder Benutzern in einem anderen Konto), um Objekte hochzuladen.

Wenn ein Bucket-Besitzer einem anderen Konto erlaubt, Objekte in einen Bucket hochzuladen, besitzt standardmäßig der Bucket-Besitzer alle Objekte, die in seinen Bucket hochgeladen wurden. Wenn jedoch sowohl die Einstellungen für den Bucket Owner als auch für den Bucket Preferred Bucket des Bucket-Besitzers deaktiviert sind, AWS-Konto derjenige, der die Objekte hochlädt, besitzt diese Objekte, und der Bucket-Besitzer hat keine Berechtigungen für die Objekte, die einem anderen Konto gehören, mit den folgenden Ausnahmen:

- Der Bucket-Eigentümer zahlt die Rechnungen. Ein Bucket-Eigentümer kann explizit den Zugriff auf Objekte verweigern oder Objekte im Bucket löschen, unabhängig davon, wem sie gehören.
- Der Bucket-Besitzer kann beliebige Objekte archivieren oder archivierte Objekte wiederherstellen, unabhängig davon, wem sie gehören. Die Archivierung bezieht sich auf die Speicherklasse, die beim Speichern der Objekte verwendet wurde. Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Tools zur Zugriffsverwaltung

Amazon S3 enthält eine Vielzahl von Sicherheitsfunktionen und -tools. Im Folgenden finden Sie eine umfassende Liste dieser Funktionen und Tools. Sie benötigen nicht alle diese Tools zur Zugriffsverwaltung, aber Sie müssen eines oder mehrere verwenden, um Zugriff auf Ihre Amazon S3 S3-Ressourcen zu gewähren. Durch die richtige Anwendung dieser Tools kann sichergestellt werden, dass Ihre Ressourcen nur den vorgesehenen Benutzern zugänglich sind.

Das am häufigsten verwendete Zugriffsverwaltungstool ist eine Zugriffsrichtlinie. Bei einer Zugriffsrichtlinie kann es sich um eine ressourcenbasierte Richtlinie handeln, die an eine angehängt ist AWS Ressource, z. B. eine Bucket-Richtlinie für einen Bucket. Eine Zugriffsrichtlinie kann auch eine identitätsbasierte Richtlinie sein, die an eine angehängt ist AWS Identity and Access Management (IAM) Identität, z. B. ein IAM Benutzer, eine Gruppe oder eine Rolle. Schreiben Sie eine Zugriffsrichtlinie, die Sie gewähren möchten AWS-Konten und IAM Benutzern, Gruppen und Rollen die Erlaubnis, Operationen an einer Ressource auszuführen. Sie können beispielsweise einer anderen Person die PUT Object Erlaubnis erteilen AWS-Konto damit das andere Konto Objekte in Ihren Bucket hochladen kann.

Eine Zugriffsrichtlinie beschreibt, wer Zugriff auf welche Dinge hat. Wenn Amazon S3 eine Anfrage erhält, muss es alle Zugriffsrichtlinien auswerten, um zu entscheiden, ob die Anfrage autorisiert oder abgelehnt werden soll. Weitere Informationen dazu, wie Amazon S3 diese Richtlinien auswertet, finden Sie unter [Wie Amazon S3 eine Anforderung autorisiert](#).

Im Folgenden sind die in Amazon S3 verfügbaren Zugriffsverwaltungstools aufgeführt.

Bucket-Richtlinie

Eine Amazon S3 S3-Bucket-Richtlinie ist ein JSON -formatiertes [AWS Identity and Access Management \(IAM\) ressourcenbasierte Richtlinie](#), die an einen bestimmten Bucket angehängt ist. Verwenden Sie Bucket-Richtlinien, um andere zu gewähren AWS-Konten oder IAM identifiziert Berechtigungen für den Bucket und die darin enthaltenen Objekte. Viele Anwendungsfälle für die S3-Zugriffsverwaltung können durch die Verwendung einer Bucket-Richtlinie erfüllt werden. Mit Bucket-Richtlinien können Sie den Bucket-Zugriff personalisieren, um sicherzustellen, dass nur die von Ihnen genehmigten Identitäten auf Ressourcen zugreifen und darin Aktionen ausführen können. Weitere Informationen finden Sie unter [Bucket-Richtlinien für Amazon S3](#).

Hier finden Sie ein Beispiel für eine Bucket-Richtlinie. Sie drücken die Bucket-Richtlinie mithilfe einer JSON Datei aus. Diese Beispielrichtlinie gewährt einer IAM Rolle Leseberechtigung für alle Objekte im Bucket. Sie enthält eine Anweisung mit dem Namen `BucketLevelReadPermissions`, die die `s3:GetObject` Aktion (Leseberechtigung) für Objekte in einem Bucket mit dem Namen `amzn-s3-demo-bucket1`. Durch die `Principal` Angabe einer IAM Rolle als „gewährt diese Richtlinie jedem IAM Benutzer mit dieser Rolle Zugriff. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BucketLevelReadPermissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-role"
      },
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket1/*"]
    }
  ]
}
```

Note

Vermeiden Sie beim Erstellen von Richtlinien die Verwendung von Platzhalterzeichen (*) im `Principal`-Element, da ein Platzhalterzeichen allen Benutzer den Zugriff auf Ihre Amazon-S3-Ressourcen ermöglicht. Führen Sie stattdessen explizit Benutzer oder Gruppen auf, die auf den Bucket zugreifen dürfen, oder führen Sie mithilfe einer Bedingungsklausel in der Richtlinie Bedingungen auf, die erfüllt werden müssen. Erteilen Sie Ihren Benutzern oder

Gruppen außerdem gegebenenfalls spezifische Berechtigungen, anstatt Platzhalterzeichen für die Aktionen zu verwenden.

Identitätsbasierte Richtlinien

Eine identitätsbasierte Richtlinie oder IAM Benutzerrichtlinie ist eine Art von [AWS Identity and Access Management \(IAM\) Richtlinie](#). Eine identitätsbasierte Richtlinie ist eine JSON -formatierte Richtlinie, die IAM Benutzern, Gruppen oder Rollen in Ihrem AWS Konto. Sie können identitätsbasierte Richtlinien verwenden, um einer IAM Identität Zugriff auf Ihre Buckets oder Objekte zu gewähren. Sie können IAM Benutzer, Gruppen und Rollen in Ihrem Konto erstellen und ihnen Zugriffsrichtlinien zuordnen. Anschließend können Sie Zugriff gewähren auf AWS Ressourcen, einschließlich Amazon S3 S3-Ressourcen. Weitere Informationen finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#).

Im Folgenden finden Sie ein Beispiel für eine identitätsbasierte Richtlinie. Die Beispielrichtlinie ermöglicht es der zugehörigen IAM Rolle, sechs verschiedene Amazon S3 S3-Aktionen (Berechtigungen) für einen Bucket und die darin enthaltenen Objekte auszuführen. Wenn Sie diese Richtlinie einer IAM Rolle in Ihrem Konto zuordnen und die Rolle einigen Ihrer IAM Benutzer zuweisen, können die Benutzer mit dieser Rolle diese Aktionen mit den in Ihrer Richtlinie angegebenen Ressourcen (Buckets) ausführen. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AssignARoleActions",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1/*",
        "arn:aws:s3:::amzn-s3-demo-bucket1"
      ]
    }
  ]
}
```

```
    },  
    {  
      "Sid": "AssignARoleActions2",  
      "Effect": "Allow",  
      "Action": "s3:ListAllMyBuckets",  
      "Resource": "*"   
    }  
  ]  
}
```

S3 Access Grants

Verwenden Sie S3 Access Grants, um Zugriffsberechtigungen für Ihre Amazon S3 S3-Daten für beide Identitäten in Unternehmensidentitätsverzeichnissen zu erstellen, z. B. Active Directory, und zu AWS Identity and Access Management (IAM) Identitäten. S3 Access Grants hilft Ihnen dabei, Datenberechtigungen in großem Umfang zu verwalten. Darüber hinaus protokolliert S3 Access Grants die Identität des Endbenutzers und die Anwendung, die für den Zugriff auf die S3-Daten verwendet wurde AWS CloudTrail. Dies bietet einen detaillierten Auditverlauf bis hin zur Endbenutzeridentität für den gesamten Zugriff auf die Daten in Ihren S3-Buckets. Weitere Informationen finden Sie unter [Verwalten des Zugriffs mit S3-Zugriffsberechtigungen](#).

Zugriffspunkte

Amazon S3 Access Points vereinfacht die Verwaltung des Datenzugriffs in großem Umfang für Anwendungen, die gemeinsam genutzte Datensätze auf S3 verwenden. Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. Sie können Access Points verwenden, um S3-Objektoperationen in großem Umfang durchzuführen, wie z. B. das Hochladen und Abrufen von Objekten. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um den Zugriff auf Ihre S3-Objekte detailliert zu kontrollieren. S3 Access Points können Buckets im selben Konto oder in einem anderen vertrauenswürdigen Konto zugeordnet werden. Bei Access Point-Richtlinien handelt es sich um ressourcenbasierte Richtlinien, die in Verbindung mit der zugrunde liegenden Bucket-Richtlinie bewertet werden. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Liste der Zugriffskontrollen () ACL

An ACL ist eine Liste von Zuschüssen, in der der Empfänger und die erteilte Genehmigung identifiziert werden. ACLs anderen Personen grundlegende Lese- oder Schreibberechtigungen gewähren AWS-Konten. ACLs verwenden Sie ein Amazon S3 S3-spezifisches SchemaXML. An

ACL ist ein Typ von [AWS Identity and Access Management \(IAM\) Richtlinie](#). Ein Objekt ACL wird verwendet, um den Zugriff auf ein Objekt zu verwalten, und ein Bucket ACL wird verwendet, um den Zugriff auf einen Bucket zu verwalten. Bei Bucket-Richtlinien gibt es eine einzige Richtlinie für den gesamten Bucket, aber Objekte ACLs werden für jedes Objekt angegeben. Es wird empfohlen, die Option ACLs ausgeschaltet zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen zur Verwendung finden ACLs Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Warning

Die meisten modernen Anwendungsfälle in Amazon S3 erfordern nicht die Verwendung von ACLs.

Im Folgenden finden Sie ein Beispiel für einen Bucket ACL. Der Grant im ACL zeigt einen Bucket-Besitzer, der über Vollzugriff verfügt.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>Owner-Canonical-User-ID</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Canonical
User">
        <ID>Owner-Canonical-User-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Eigentum am Objekt

Um den Zugriff auf Ihre Objekte zu verwalten, müssen Sie der Eigentümer des Objekts sein. Sie können die Einstellung „Objekteigentum“ auf Bucketebene verwenden, um den Besitz von Objekten zu kontrollieren, die in Ihren Bucket hochgeladen wurden. Verwenden Sie auch „Objekteigentum“, um

die Option zu aktivieren. ACLs Standardmäßig ist für Object Ownership die Einstellung Bucket Owner enforced festgelegt und alle Optionen ACLs sind deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet ausschließlich den Zugriff auf Daten. Um den Zugriff zu verwalten, verwendet der Bucket-Besitzer Richtlinien oder ein anderes Zugriffsverwaltungstool, ausgenommen ACLs. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Object Ownership verfügt über drei Einstellungen, mit denen Sie sowohl den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu aktivieren ACLs:

ACLs ausgeschaltet

- Bucket-Besitzer erzwungen (Standard) — ACLs sind ausgeschaltet, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLs wirken sich nicht auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

ACLs eingeschaltet

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreiben ACL.
- Objekt-Writer — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Weitere bewährte Methoden

Erwägen Sie die Verwendung der folgenden Bucket-Einstellungen und Tools, um Daten während der Übertragung und Speicherung zu schützen. Beide sind entscheidend für die Aufrechterhaltung der Integrität und Zugänglichkeit Ihrer Daten:

- Öffentlichen Zugriff blockieren — Deaktivieren Sie nicht die Standardeinstellung „Öffentlichen Zugriff blockieren“ auf Bucket-Ebene. Diese Einstellung blockiert standardmäßig den öffentlichen Zugriff auf Ihre Daten. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).
- S3-Versionierung — Aus Gründen der Datenintegrität können Sie die S3-Versioning-Bucket-Einstellung implementieren, mit der Ihre Objekte bei Aktualisierungen versioniert werden, anstatt sie zu überschreiben. Sie können S3 Versioning verwenden, um bei Bedarf eine frühere Version

beizubehalten, abzurufen und wiederherzustellen. Weitere Informationen über S3 Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

- S3 Object Lock — S3 Object Lock ist eine weitere Einstellung, die Sie implementieren können, um Datenintegrität zu erreichen. Diese Funktion kann ein write-once-read-many (WORM) -Modell implementieren, um Objekte unveränderlich zu speichern. Weitere Informationen über die Objektsperre finden Sie unter [Verwenden der S3-Objektsperre](#).
- Objektverschlüsselung — Amazon S3 bietet mehrere Objektverschlüsselungsoptionen, die Daten während der Übertragung und Speicherung schützen. Die serverseitige Verschlüsselung verschlüsselt Ihr Objekt, bevor es auf Festplatten in seinen Rechenzentren gespeichert wird, und entschlüsselt es dann, wenn Sie die Objekte herunterladen. Wenn Sie Ihre Anfrage authentifizieren und über Zugriffsberechtigungen verfügen, gibt es keinen Unterschied in der Art und Weise, wie Sie auf verschlüsselte oder unverschlüsselte Objekte zugreifen. Weitere Informationen finden Sie unter [Schützen von Daten mit serverseitiger Verschlüsselung](#). S3 verschlüsselt neu hochgeladene Objekte standardmäßig. Weitere Informationen finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#). Client-seitige Verschlüsselung ist der Vorgang, bei dem Daten verschlüsselt werden, bevor sie an Amazon S3 gesendet werden. Weitere Informationen finden Sie unter [Schützen von Daten mithilfe der clientseitigen Verschlüsselung](#).
- Signaturmethoden — Signature Version 4 ist der Prozess des Hinzufügens von Authentifizierungsinformationen AWS Anfragen gesendet vonHTTP. Aus Sicherheitsgründen richten sich die meisten Anfragen an AWS müssen mit einem Zugriffsschlüssel signiert sein, der aus einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel besteht. Diese beiden Schlüssel werden in der Regel als Sicherheitsanmeldeinformationen bezeichnet. Weitere Informationen finden Sie unter [Authentifizieren von Anfragen \(AWSSignaturprozess für Signature Version 4 und Signature Version 4\)](#).

Aktionen

Eine vollständige Liste der S3-Berechtigungen und Bedingungsschlüssel finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Aktionen

Das Tool AWS Identity and Access Management (IAM) Aktionen für Amazon S3 sind die möglichen Aktionen, die für einen S3-Bucket oder ein S3-Objekt ausgeführt werden können. Sie weisen diese

Aktionen Identitäten zu, damit sie auf Ihre S3-Ressourcen einwirken können. Beispiele für S3-Aktionen sind `s3:GetObject` das Lesen von Objekten in einem Bucket und `s3:PutObject` das Schreiben von Objekten in einen Bucket.

Bedingungsschlüssel

Zusätzlich zu Aktionen sind IAM Bedingungsschlüssel darauf beschränkt, Zugriff nur dann zu gewähren, wenn eine Bedingung erfüllt ist. Bedingungsschlüssel sind optional.

Note

In einer ressourcenbasierten Zugriffsrichtlinie, z. B. einer Bucket-Richtlinie, oder in einer identitätsbasierten Richtlinie können Sie Folgendes angeben:

- Eine Aktion oder eine Reihe von Aktionen im `Action` Element der Grundsatzerklärung.
- Im `Effect` Element der Grundsatzerklärung können Sie angeben, ob die aufgeführten Aktionen gewährt werden `Allow` sollen, oder Sie können angeben, `Deny` dass die aufgelisteten Aktionen blockiert werden sollen. Um die Praxis der geringsten Rechte weiter aufrechtzuerhalten, sollten die `Deny` Aussagen im `Effect` Element der Zugangsrichtlinie so weit wie möglich und die `Allow` Aussagen so eng wie möglich sein. `Deny` Die mit der `s3:*` Maßnahme verbundenen Effekte sind eine weitere gute Möglichkeit, bewährte Verfahren für die Zustimmung zu den Identitäten umzusetzen, die in den Erklärungen zu den politischen Bedingungen enthalten sind.
- Ein Bedingungsschlüssel im `Condition` Element einer Grundsatzerklärung.

Anwendungsfälle für die Zugriffsverwaltung

Amazon S3 bietet Ressourcenbesitzern eine Vielzahl von Tools, mit denen sie Zugriff gewähren können. Welches S3-Zugriffsverwaltungstool Sie verwenden, hängt von den S3-Ressourcen ab, die Sie teilen möchten, den Identitäten, denen Sie Zugriff gewähren, und den Aktionen, die Sie zulassen oder verweigern möchten. Möglicherweise möchten Sie eines oder eine Kombination von S3-Zugriffsverwaltungstools verwenden, um den Zugriff auf Ihre S3-Ressourcen zu verwalten.

In den meisten Fällen können Sie eine Zugriffsrichtlinie verwenden, um Berechtigungen zu verwalten. Eine Zugriffsrichtlinie kann eine ressourcenbasierte Richtlinie sein, die an eine Ressource, z. B. einen Bucket, oder eine andere Amazon S3 S3-Ressource ([S3-Ressourcen](#)) angehängt ist. Eine

Zugriffsrichtlinie kann auch eine identitätsbasierte Richtlinie sein, die an eine angehängt ist AWS Identity and Access Management (IAM) Benutzer, Gruppe oder Rolle in Ihrem Konto. Möglicherweise stellen Sie fest, dass eine Bucket-Richtlinie für Ihren Anwendungsfall besser geeignet ist. Weitere Informationen finden Sie unter [Bucket-Richtlinien für Amazon S3](#). Alternativ mit AWS Identity and Access Management (IAM) können Sie IAM Benutzer, Gruppen und Rollen in Ihrem erstellen AWS-Konto und ihren Zugriff auf Buckets und Objekte mithilfe identitätsbasierter Richtlinien verwalten. Weitere Informationen finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#).

Um Ihnen den Umgang mit diesen Zugriffsverwaltungsoptionen zu erleichtern, finden Sie im Folgenden allgemeine Anwendungsfälle und Empfehlungen für die einzelnen S3-Zugriffsverwaltungstools von Amazon S3 S3-Kunden.

Das Tool AWS-Konto Der Besitzer möchte Buckets nur mit Benutzern innerhalb desselben Kontos teilen

Alle Tools zur Zugriffsverwaltung können diesen grundlegenden Anwendungsfall erfüllen. Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- **Bucket-Richtlinie** — Wenn Sie Zugriff auf einen Bucket oder eine kleine Anzahl von Buckets gewähren möchten oder wenn Ihre Bucket-Zugriffsberechtigungen von Bucket zu Bucket ähnlich sind, verwenden Sie eine Bucket-Richtlinie. Mit Bucket-Richtlinien verwalten Sie eine Richtlinie für jeden Bucket. Weitere Informationen finden Sie unter [Bucket-Richtlinien für Amazon S3](#).
- **Identitätsbasierte Richtlinie** — Wenn Sie eine sehr große Anzahl von Buckets mit unterschiedlichen Zugriffsberechtigungen für jeden Bucket haben und nur wenige Benutzerrollen verwalten müssen, können Sie eine IAM Richtlinie für Benutzer, Gruppen oder Rollen verwenden. IAMRichtlinien sind auch eine gute Option, wenn Sie den Benutzerzugriff auf andere verwalten AWS Ressourcen sowie Amazon S3 S3-Ressourcen. Weitere Informationen finden Sie unter [Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen](#).
- **S3 Access Grants** — Sie können S3 Access Grants verwenden, um Zugriff auf Ihre S3-Buckets, -Präfixe oder Objekte zu gewähren. Mit S3 Access Grants können Sie unterschiedliche Berechtigungen auf Objektebene nach Bedarf festlegen. Bucket-Richtlinien sind dagegen auf eine Größe von 20 KB begrenzt. Weitere Informationen finden Sie unter [Erste Schritte mit S3 Access Grants](#).
- **Access Points** — Sie können Access Points verwenden, bei denen es sich um so genannte Netzwerkendpunkte handelt, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um Ihnen eine detaillierte Kontrolle über

den Zugriff auf Ihre S3-Objekte zu geben. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Das Tool AWS-Konto Der Besitzer möchte Buckets oder Objekte mit Benutzern eines anderen Benutzers teilen AWS Konto (kontoübergreifend)

Um einem anderen die Erlaubnis zu erteilen AWS-Konto, müssen Sie eine Bucket-Richtlinie oder eines der folgenden empfohlenen Zugriffsverwaltungstools verwenden. Für diesen Anwendungsfall können Sie keine identitätsbasierte Zugriffsrichtlinie verwenden. Weitere Informationen zur Gewährung von kontenübergreifendem Zugriff finden Sie unter [Wie stelle ich kontenübergreifenden Zugriff auf Objekte bereit, die sich in Amazon S3 S3-Buckets befinden?](#)

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Bucket-Richtlinie — Mit Bucket-Richtlinien verwalten Sie eine Richtlinie für jeden Bucket. Weitere Informationen finden Sie unter [Bucket-Richtlinien für Amazon S3](#).
- S3 Access Grants — Sie können S3 Access Grants verwenden, um kontoübergreifende Berechtigungen für Ihre S3-Buckets, -Präfixe oder Objekte zu gewähren. Sie können S3 Access Grants verwenden, um unterschiedliche Berechtigungen auf Objektebene nach Bedarf festzulegen. Bucket-Richtlinien sind dagegen auf eine Größe von 20 KB begrenzt. Weitere Informationen finden Sie unter [Erste Schritte mit S3 Access Grants](#).
- Access Points — Sie können Access Points verwenden, bei denen es sich um so genannte Netzwerkendpunkte handelt, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um Ihnen eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu geben. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Das Tool AWS-Konto Besitzer oder Bucket-Besitzer müssen Berechtigungen auf Objekt- oder Präfix-Ebene gewähren, und diese Berechtigungen variieren von Objekt zu Objekt oder von Präfix zu Präfix

In einer Bucket-Richtlinie können Sie beispielsweise Zugriff auf Objekte innerhalb eines Buckets gewähren, die ein bestimmtes [Schlüsselnamenpräfix](#) oder ein bestimmtes Tag haben. Sie können Leseberechtigungen für Objekte gewähren, die mit dem Schlüsselnamenpräfix beginnenlogs/. Wenn Ihre Zugriffsberechtigungen jedoch je nach Objekt variieren, ist es möglicherweise nicht praktikabel, einzelnen Objekten mithilfe einer Bucket-Richtlinie Berechtigungen zu gewähren, zumal Bucket-Richtlinien auf eine Größe von 20 KB begrenzt sind.

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- S3 Access Grants — Sie können S3 Access Grants verwenden, um Berechtigungen auf Objekt- oder Präfix-Ebene zu verwalten. Im Gegensatz zu Bucket-Richtlinien können Sie S3 Access Grants verwenden, um unterschiedliche Berechtigungen auf Objektebene nach Bedarf festzulegen. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt. Weitere Informationen finden Sie unter [Erste Schritte mit S3 Access Grants](#).
- Access Points — Sie können Access Points verwenden, um Berechtigungen auf Objekt- oder Präfix-Ebene zu verwalten. Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu erhalten. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).
- ACLs— Wir raten davon ab, Zugriffskontrolllisten (ACLs) zu verwenden, vor allem, weil sie auf 100 Grants pro Objekt begrenzt sind. Wenn Sie sich jedoch dafür entscheiden, diese Option zu aktivieren, setzen Sie in Ihren Bucket-Einstellungen die Option Objekteigentum auf „Bevorzugter und ACLs aktivierter Bucket Owner“. Mit dieser Einstellung gehören neue Objekte, die mit dem gespeicherten Objekt `bucket-owner-full-control` geschriebenen ACL werden, automatisch dem Bucket-Besitzer und nicht dem Objektschreiber. Sie können dann Objekt verwenden ACLs, bei dem es sich um eine Zugriffsrichtlinie im XML -Format handelt, um anderen Benutzern Zugriff auf das Objekt zu gewähren. Weitere Informationen finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Das Tool AWS-Konto Der Besitzer oder Bucket-Besitzer möchte den Bucket-Zugriff nur auf ein bestimmtes Konto beschränken IDs

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Bucket-Richtlinie — Mit Bucket-Richtlinien verwalten Sie eine Richtlinie für jeden Bucket. Weitere Informationen finden Sie unter [Bucket-Richtlinien für Amazon S3](#).
- Access Points — Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu erhalten. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Das Tool AWS-Konto Der Besitzer oder Bucket-Besitzer möchte für jeden Benutzer oder jede Anwendung, die auf ihre Daten zugreift, eigene Endpunkte

Für diesen Anwendungsfall empfehlen wir das folgende Zugriffsverwaltungstool:

- **Access Points** — Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu erhalten. Jeder Zugriffspunkt erzwingt eine benutzerdefinierte Zugriffspunktrichtlinie, die in Verbindung mit der Bucket-Richtlinie funktioniert, die dem zugrunde liegenden Bucket zugeordnet ist. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Das Tool AWS-Konto Der Besitzer oder Bucket-Besitzer muss den Zugriff von Virtual Private Cloud (VPC) -Endpunkten für S3 verwalten

Virtual Private Cloud (VPC) -Endpunkte für Amazon S3 sind logische Entitäten innerhalb einer VPC, die nur Konnektivität zu S3 ermöglichen. Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- **Buckets in einer VPC Einstellung** — Sie können eine Bucket-Richtlinie verwenden, um zu kontrollieren, wer auf Ihre Buckets zugreifen darf und auf welche VPC Endpunkte sie zugreifen können. Weitere Informationen finden Sie unter [Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien](#).
- **Access Points** — Wenn Sie Access Points einrichten möchten, können Sie eine Zugriffspunkt-Richtlinie verwenden. Sie können jeden Access Point so konfigurieren, dass er nur Anfragen von einer virtuellen privaten Cloud (VPC) akzeptiert, um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk einzuschränken. Sie können auch benutzerdefinierte Block Public Access-Einstellungen für jeden Zugriffspunkt konfigurieren. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).

Das Tool AWS-Konto Der Besitzer oder Bucket-Besitzer muss eine statische Website öffentlich zugänglich machen

Mit S3 können Sie eine statische Website hosten und jedem ermöglichen, den Inhalt der Website anzusehen, die von einem S3-Bucket aus gehostet wird.

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Amazon CloudFront — Mit dieser Lösung können Sie eine statische Amazon S3 S3-Website für die Öffentlichkeit hosten und gleichzeitig den gesamten öffentlichen Zugriff auf den Inhalt eines Buckets blockieren. Wenn Sie alle vier S3 Block Public Access-Einstellungen aktiviert lassen und eine statische S3-Website hosten möchten, können Sie Amazon CloudFront Origin Access Control (OAC) verwenden. Amazon CloudFront bietet die Funktionen, die für die Einrichtung einer sicheren statischen Website erforderlich sind. Außerdem können statische Amazon S3 S3-Websites, die diese Lösung nicht verwenden, nur HTTP Endgeräte unterstützen. CloudFront verwendet den dauerhaften Speicher von Amazon S3 und bietet gleichzeitig zusätzliche Sicherheitsheader, wie HTTPS z. HTTPS erhöht die Sicherheit, indem es eine normale HTTP Anfrage verschlüsselt und vor häufigen Cyberangriffen schützt.

Weitere Informationen finden Sie unter [Erste Schritte mit einer sicheren statischen Website](#) im Amazon CloudFront Developer Guide.

- Ihren Amazon S3 S3-Bucket öffentlich zugänglich machen — Sie können einen Bucket so konfigurieren, dass er als öffentlich zugängliche statische Website verwendet wird.

Warning

Wir empfehlen diese Methode nicht. Stattdessen empfehlen wir Ihnen, statische Amazon S3 S3-Websites als Teil von Amazon zu verwenden CloudFront. Weitere Informationen finden Sie in der vorherigen Option oder unter [Erste Schritte mit einer sicheren statischen Website](#).

Um eine statische Amazon S3 S3-Website ohne Amazon CloudFront zu erstellen, müssen Sie zunächst alle Block Public Access-Einstellungen deaktivieren. Wenn Sie die Bucket-Richtlinie für Ihre statische Website schreiben, achten Sie darauf, nur `s3:GetObject`-Aktionen und keine `ListObject`- oder `PutObject`-Berechtigungen zuzulassen. Dadurch wird sichergestellt, dass Benutzer nicht alle Objekte in Ihrem Bucket anzeigen oder eigene Inhalte hinzufügen können. Weitere Informationen finden Sie unter [Festlegen von Berechtigungen für den Website-Zugriff](#).

Das Tool AWS-Konto Der Besitzer oder Bucket-Besitzer möchte den Inhalt eines Buckets öffentlich zugänglich machen

Wenn Sie einen neuen Amazon S3 S3-Bucket erstellen, ist die Einstellung Block Public Access standardmäßig aktiviert. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Wir empfehlen nicht, öffentlichen Zugriff auf Ihren Bucket zu gewähren. Wenn Sie dies jedoch für einen bestimmten Anwendungsfall tun müssen, empfehlen wir das folgende Zugriffsverwaltungstool für diesen Anwendungsfall:

- Einstellung „Öffentlichen Zugriff blockieren“ deaktivieren — Ein Bucket-Besitzer kann nicht authentifizierte Anfragen an den Bucket zulassen. Nicht authentifizierte [PUTObjektanforderungen](#) sind beispielsweise zulässig, wenn für einen Bucket eine öffentliche Bucket-Richtlinie gilt oder wenn ein Bucket öffentlichen Zugriff gewährt. ACL Alle nicht authentifizierte Anfragen werden von anderen willkürlich gestellt AWS Benutzer oder sogar nicht authentifizierte, anonyme Benutzer. Dieser Benutzer wird ACLs durch die spezifische kanonische Benutzer-ID repräsentiert. 65a011a29cdf8ec533ec3d1c3caae921c Wenn ein Objekt in ein WRITE oder hochgeladen wird FULL_CONTROL, gewährt dies ausdrücklich Zugriff auf die Gruppe „Alle Benutzer“ oder den anonymen Benutzer. Weitere Informationen zu öffentlichen Bucket-Richtlinien und öffentlichen Zugriffskontrolllisten (ACLs) finden Sie unter [Die Bedeutung von „öffentlich“](#).

Das Tool AWS-Konto Der Besitzer oder Bucket-Besitzer hat die Größenbeschränkungen der Zugriffsrichtlinie überschritten

Sowohl für Bucket-Richtlinien als auch für identitätsbasierte Richtlinien gilt eine Größenbeschränkung von 20 KB. Wenn Ihre Anforderungen an Zugriffsberechtigungen komplex sind, überschreiten Sie möglicherweise diese Größenbeschränkung.

Wir haben für diesen Anwendungsfall die folgenden Tools zur Zugriffsverwaltung empfohlen:

- Access Points — Verwenden Sie Access Points, wenn dies für Ihren Anwendungsfall funktioniert. Bei Access Points hat jeder Bucket mehrere benannte Netzwerkendpunkte, von denen jeder seine eigene Zugriffspunktrichtlinie hat, die mit der zugrundeliegenden Bucket-Richtlinie funktioniert. Access Points können jedoch nur auf Objekte, nicht auf Buckets, einwirken und unterstützen keine regionsübergreifende Replikation. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon S3-Zugangspunkten](#).
- S3 Access Grants — Verwenden Sie S3 Access Grants, das eine sehr große Anzahl von Zugriffen unterstützt, die Zugriff auf Buckets, Präfixe oder Objekte gewähren. Weitere Informationen finden Sie unter [Erste Schritte mit S3 Access Grants](#).

Das Tool AWS-Konto Die Inhaber- oder Administratorrolle möchte Benutzern oder Gruppen in einem Unternehmensverzeichnis direkt Bucket-, Präfix- oder Objektzugriff gewähren

Anstatt Benutzer, Gruppen und Rollen über zu verwalten AWS Identity and Access Management (IAM) können Sie Ihr Unternehmensverzeichnis hinzufügen AWS IAM Identity Center. Weitere Informationen finden Sie unter [Was ist IAM Identity Center?](#) .

Nachdem Sie Ihr Unternehmensverzeichnis hinzugefügt haben AWS IAM Identity Center, empfehlen wir Ihnen, das folgende Zugriffsverwaltungstool zu verwenden, um den Identitäten des Unternehmensverzeichnisses Zugriff auf Ihre S3-Ressourcen zu gewähren:

- S3 Access Grants — Verwenden Sie S3 Access Grants, das die Gewährung von Zugriff auf Benutzer oder Rollen in Ihrem Unternehmensverzeichnis unterstützt. Weitere Informationen finden Sie unter [Erste Schritte mit S3 Access Grants](#).

Das Tool AWS-Konto Der Besitzer oder Bucket-Besitzer möchte das geben AWS CloudFront Dienstzugriff, um CloudFront Logs in einen S3-Bucket zu schreiben

Wir haben das folgende Zugriffsverwaltungstool für diesen Anwendungsfall empfohlen:

- Bucket ACL — Der einzig empfohlene Anwendungsfall für Bucket ACLs ist die Gewährung von Berechtigungen für bestimmte AWS-Services, wie das CloudFront `awslogsdelivery` Amazon-Konto. Wenn Sie eine Distribution erstellen oder aktualisieren und die CloudFront Protokollierung aktivieren, wird der Bucket CloudFront aktualisiert, ACL sodass das `awslogsdelivery` Konto `FULL_CONTROL` berechtigt ist, Logs in Ihren Bucket zu schreiben. Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Konfiguration der Standardprotokollierung und für den Zugriff auf Ihre Protokolldateien](#) im Amazon CloudFront Developer Guide. Wenn der Bucket, in dem die Logs gespeichert werden, die Einstellung Bucket Ownerforced für die Deaktivierung von S3 Object Ownership verwendetACLs, CloudFront können keine Logs in den Bucket geschrieben werden. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Sie als Bucket-Besitzer möchten die volle Kontrolle über Objekte behalten, die dem Bucket von anderen Benutzern hinzugefügt werden

Sie können anderen Konten Zugriff gewähren, um Objekte in Ihren Bucket hochzuladen, indem Sie eine Bucket-Richtlinie, einen Access Point oder S3 Access Grants verwenden. Wenn Sie

kontoübergreifenden Zugriff auf Ihren Bucket gewährt haben, können Sie sicherstellen, dass Sie weiterhin die volle Kontrolle über alle in Ihren Bucket hochgeladenen Objekte haben.

Für diesen Anwendungsfall haben wir das folgende Zugriffsverwaltungstool empfohlen:

- Objekteigentum — Behalten Sie für die Einstellung Object Ownership auf Bucket-Ebene die Standardeinstellung Bucket Ownerforced bei.

Fehlerbehebung bei der Zugriffsverwaltung

Die folgenden Ressourcen können Ihnen bei der Behebung von Problemen mit der S3-Zugriffsverwaltung helfen:

Beheben von Fehlern aufgrund einer Zugriffsverweigerung (403 Forbidden)

Wenn Sie auf Probleme mit der Zugriffsverweigerung stoßen, überprüfen Sie die Einstellungen auf Konto- und Bucket-Ebene. Überprüfen Sie außerdem die Zugriffsverwaltungsfunktion, mit der Sie den Zugriff gewähren, um sicherzustellen, dass die Richtlinie, Einstellung oder Konfiguration korrekt ist. Weitere Informationen zu den häufigsten Ursachen für den Fehler „Zugriff verweigert“ (403 Forbidden) in Amazon S3 finden Sie unter [Problembehandlung bei Zugriff verweigert \(403 Forbidden\) Fehler in Amazon S3](#).

IAMAccess Analyzer für S3

Wenn Sie keine Ihrer Ressourcen öffentlich zugänglich machen oder den öffentlichen Zugriff auf Ihre Ressourcen einschränken möchten, können Sie IAM Access Analyzer für S3 verwenden. Verwenden Sie IAM Access Analyzer for S3 auf der Amazon S3-Konsole, um alle Buckets zu überprüfen, die über Bucket-Zugriffskontrolllisten (ACLs), Bucket-Richtlinien oder Access Point-Richtlinien verfügen, die öffentlichen oder gemeinsamen Zugriff gewähren. IAMAccess Analyzer for S3 warnt Sie vor Buckets, die so konfiguriert sind, dass sie allen Personen im Internet oder auf anderen Websites Zugriff gewähren AWS-Konten, einschließlich AWS-Konten außerhalb Ihrer Organisation. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Ergebnisse, die die Quelle und die Ebene des öffentlichen oder freigegebenen Zugriffs melden.

In IAM Access Analyzer for S3 können Sie den gesamten öffentlichen Zugriff auf einen Bucket mit einer einzigen Aktion blockieren. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren, es sei denn, Sie benötigen öffentlichen Zugriff, um einen bestimmten Anwendungsfall zu unterstützen. Bevor Sie den gesamten öffentlichen Zugriff blockieren, stellen Sie

sicher, dass Ihre Anwendungen auch ohne öffentlichen Zugriff ordnungsgemäß funktionieren. Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Sie können auch Ihre Berechtigungseinstellungen auf Bucket-Ebene überprüfen, um detaillierte Zugriffsebenen zu konfigurieren. Für bestimmte und geprüfte Anwendungsfälle, die öffentlichen oder freigegebenen Zugriff erfordern, können Sie Ihre Absicht bestätigen und aufzeichnen, dass der Bucket öffentlich oder freigegeben bleibt, indem Sie die Ergebnisse für den Bucket archivieren. Diese Bucket-Konfigurationen sind jederzeit aufrufbar und änderbar. Sie können Ihre Ergebnisse zu Prüfungszwecken auch als CSV Bericht herunterladen.

IAM Access Analyzer for S3 ist ohne zusätzliche Kosten auf der Amazon S3 S3-Konsole verfügbar. IAM Access Analyzer for S3 wird unterstützt von AWS Identity and Access Management (IAM) IAM Access Analyzer. Um IAM Access Analyzer for S3 auf der Amazon S3 S3-Konsole zu verwenden, müssen Sie die [IAM-Konsole aufrufen](#) und in IAM Access Analyzer für jede einzelne Region einen Analyzer auf Kontoebene erstellen.

Weitere Informationen zu IAM Access Analyzer for S3 finden Sie unter [Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3](#)

Protokollierung und Überwachung

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Amazon S3 S3-Lösungen, sodass Sie einen Zugriffsfehler einfacher debuggen können. Die Protokollierung kann Einsichten in Fehler bereitstellen, die Benutzer erhalten. Sie erhalten auch Informationen, wann welche Anforderungen übermittelt werden. AWS bietet verschiedene Tools zur Überwachung Ihrer Amazon S3 S3-Ressourcen, wie z. B. die folgenden:

- AWS CloudTrail
- Amazon-S3-Zugriffsprotokolle
- AWS Trusted Advisor
- Amazon CloudWatch

Weitere Informationen finden Sie unter [Protokollierung und Überwachung in Amazon S3](#).

Themen

- [Identity and Access Management für Amazon S3](#)
- [Verwalten des Zugriffs mit S3-Zugriffsberechtigungen](#)
- [Verwaltung des Zugriffs mit ACLs](#)

- [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#)
- [Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3](#)
- [Überprüfen der Bucket-Eigentümerschaft mit Bucket-Eigentümer-Bedingung](#)
- [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#)

Identity and Access Management für Amazon S3

AWS Identity and Access Management (IAM) ist ein AWS-Service das hilft einem Administrator, den Zugriff auf sicher zu kontrollieren AWS Ressourcen schützen. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon S3 S3-Ressourcen zu verwenden. IAM ist ein AWS-Service das Sie ohne zusätzliche Kosten nutzen können.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert Amazon S3 mit IAM](#)
- [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#)
- [Richtlinien und Berechtigungen in Amazon S3](#)
- [Bucket-Richtlinien für Amazon S3](#)
- [Identitätsbasierte Richtlinien für Amazon S3](#)
- [Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten](#)
- [Wie Amazon S3 eine Anforderung autorisiert](#)

- [AWS verwaltete Richtlinien für Amazon S3](#)
- [Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens](#)
- [Problembehandlung Amazon S3 S3-Identität und -Zugriff](#)

Zielgruppe

Wie verwenden Sie AWS Identity and Access Management (IAM) unterscheidet sich je nach der Arbeit, die Sie in Amazon S3 ausführen.

Servicebenutzer — Wenn Sie den Amazon S3 S3-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Amazon S3 S3-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie auf eine Funktion in Amazon S3 nicht zugreifen können, finden Sie weitere Informationen unter [Problembehandlung Amazon S3 S3-Identität und -Zugriff](#).

Service-Administrator — Wenn Sie in Ihrem Unternehmen für Amazon S3-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon S3. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon S3 Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die Berechtigungen Ihrer Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehen IAM. Weitere Informationen darüber, wie Ihr Unternehmen Amazon S3 nutzen IAM kann, finden Sie unter [So funktioniert Amazon S3 mit IAM](#).

IAM Administrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf Amazon S3 zu verwalten. Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien, die Sie in verwenden können IAM, finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#)

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich anmelden AWS mit Ihren Identitätsdaten. Sie müssen authentifiziert (angemeldet) sein AWS) als Root-Benutzer des AWS-Kontos, als IAM Benutzer oder durch Übernahme einer IAM Rolle.

Sie können sich anmelden bei AWS als föderierte Identität mithilfe von Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -

Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie darauf zugreifen AWS Wenn Sie den Verbund verwenden, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der anmelden AWS Management Console oder das AWS Zugangsportal. Weitere Informationen zur Anmeldung bei AWS, siehe [So melden Sie sich bei Ihrem an AWS-Konto](#) in der AWS-Anmeldung Benutzerleitfaden.

Wenn Sie darauf zugreifen AWS programmatisch AWS stellt ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie es nicht verwenden AWS Tools, Sie müssen Anfragen selbst unterschreiben. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie unter [Signieren AWS APIAnfragen](#) im IAMBenutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. Zum Beispiel AWS empfiehlt, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center Benutzerhandbuch und [Verwendung der Multi-Faktor-Authentifizierung \(\) MFA in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie eine erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle hat AWS-Services und Ressourcen im Konto. Diese Identität wird als AWS-Konto Root-Benutzer. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter [Aufgaben, für die Root-Benutzeranmeldedaten erforderlich](#) sind. IAM

Verbundidentität

Es hat sich bewährt, menschlichen Benutzern, einschließlich Benutzern, die Administratorzugriff benötigen, vorzuschreiben, für den Zugriff den Verbund mit einem Identitätsanbieter zu verwenden AWS-Services mithilfe temporärer Anmeldeinformationen.

Eine föderierte Identität ist ein Benutzer aus dem Benutzerverzeichnis Ihres Unternehmens, ein Web-Identitätsanbieter, AWS Directory Service, das Identity Center-Verzeichnis oder ein beliebiger Benutzer, der zugreift AWS-Services mithilfe von Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für eine zentralisierte Zugriffsverwaltung empfehlen wir die Verwendung AWS IAM Identity Center. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten und Anwendungen. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) in der AWS IAM Identity Center Benutzerleitfaden.

IAM-Benutzer und -Gruppen

Ein [IAMBenutzer](#) ist eine Identität innerhalb Ihres AWS-Konto das über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind](#).

Eine [IAMGruppe](#) ist eine Identität, die eine Sammlung von IAM Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Wann sollte ein IAM Benutzer \(statt einer Rolle\) erstellt werden?](#) im IAMBenutzerhandbuch.

IAMRollen

Eine [IAMRolle](#) ist eine Identität in deinem AWS-Konto das hat spezifische Berechtigungen. Es ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Sie können vorübergehend

eine IAM Rolle in der übernehmen AWS Management Console indem Sie die [Rollen wechseln](#). Sie können eine Rolle übernehmen, indem Sie einen anrufen AWS CLI or AWS API Operation oder mithilfe eines benutzerdefinierten URL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie [unter Verwenden von IAM Rollen](#) im IAM Benutzerhandbuch.

IAM Rollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAM Benutzerhandbuch unter [Erstellen einer Rolle für einen externen Identitätsanbieter](#). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center Benutzerleitfaden.
- **Temporäre IAM Benutzerberechtigungen** — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- **Kontoübergreifender Zugriff** — Sie können eine IAM Rolle verwenden, um einer Person (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Allerdings mit einigen AWS-Services, Sie können eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie [IAM im Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM
- **Serviceübergreifender Zugriff** — Einige AWS-Services Funktionen in anderen verwenden AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Zugriffssitzungen weiterleiten (FAS)** — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen in AWS, Sie gelten als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Rechte des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage AWS-Service um Anfragen an nachgelagerte Dienste zu stellen. FAS Anfragen

werden nur gestellt, wenn ein Dienst eine Anfrage erhält, die Interaktionen mit anderen erfordert AWS-Services oder zu vervollständigende Ressourcen. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

- **Servicerolle** — Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an ein AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstverknüpfte Rolle** — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen durchzuführen. Mit Diensten verknüpfte Rollen erscheinen in Ihrem AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen, die auf Amazon laufen EC2** — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS CLI or AWS APIAnfragen. Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instanz vorzuziehen. Um eine zuzuweisen AWS Sie erstellen ein EC2 Instanzprofil, das an die Instanz angehängt ist. Sie müssen einer Instanz eine Rolle zuweisen und sie allen ihren Anwendungen zur Verfügung stellen. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt](#) werden.

Informationen darüber, ob Sie IAM Rollen oder IAM Benutzer verwenden sollten, finden [Sie im Benutzerhandbuch unter Wann sollte eine IAM Rolle \(anstelle eines IAM Benutzers\) erstellt](#) werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff in AWS indem Sie Richtlinien erstellen und diese anhängen AWS Identitäten oder Ressourcen. Eine Richtlinie ist ein Objekt in AWS das, wenn es mit einer Identität oder Ressource verknüpft ist, ihre Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Principal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien sind gespeichert in AWS als JSON Dokumente. Weitere Informationen zur Struktur und zum Inhalt

von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter [Überblick über JSON Richtlinien](#).

Administratoren können verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAMRichtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der abrufen AWS Management Console, der AWS CLI, oder der AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAMRichtlinien erstellen im Benutzerhandbuch](#). IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie wählen können, finden Sie im IAMBenutzerhandbuch unter [Auswahl zwischen verwalteten Richtlinien und Inline-Richtlinien](#).

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche

Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können nicht verwenden AWS verwaltete Richtlinien aus IAM einer ressourcenbasierten Richtlinie.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3, AWS WAF, und VPC sind Beispiele für Dienste, die unterstützen ACLs. Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAM Benutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAM Benutzerhandbuch unter [Berechtigungsgrenzen für IAM Entitäten](#).
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS-Konten den Ihr Unternehmen besitzt. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos. Weitere Informationen

zu Organizations und finden Sie SCPs unter [Richtlinien zur Servicesteuerung](#) in der AWS Organizations Benutzerleitfaden.

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Sitzungsrichtlinien](#).

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Um zu erfahren, wie AWS bestimmt, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, siehe [Bewertungslogik für Richtlinien](#) im IAMBenutzerhandbuch.

So funktioniert Amazon S3 mit IAM

Informieren Sie sich vor der Verwendung IAM zur Verwaltung des Zugriffs auf Amazon S3 darüber, welche IAM Funktionen für Amazon S3 verfügbar sind.

IAMFunktionen, die Sie mit Amazon S3 verwenden können

IAMFunktion	Amazon-S3-Support
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Ja

IAMFunktion	Amazon-S3-Support
ABAC(Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Zugriffssitzungen weiterleiten (FAS)	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Teilweise

Um einen allgemeinen Überblick darüber zu erhalten, wie Amazon S3 und andere AWS Dienste funktionieren mit den meisten IAM Funktionen, siehe [AWS Dienste, mit denen IAM](#) im IAMBenutzerhandbuch gearbeitet werden kann.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Identitätsbasierte Richtlinien für Amazon S3

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAMRichtlinien erstellen im Benutzerhandbuch](#). IAM

Mit IAM identitätsbasierten Richtlinien können Sie zulässige oder verweigernde Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie im IAMBenutzerhandbuch unter [Referenz zu IAM JSON Richtlinienelementen](#).

Beispiele für identitätsbasierte Richtlinien für Amazon S3

Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#)

Ressourcenbasierte Richtlinien in Amazon S3

Unterstützt ressourcenbasierte Richtlinien: Ja

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services.

Um den kontoübergreifenden Zugriff zu ermöglichen, können Sie in einer ressourcenbasierten Richtlinie ein ganzes Konto oder IAM Entitäten in einem anderen Konto als Prinzipal angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Erlaubnis erteilen, auf die Ressource zuzugreifen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie [IAMim IAMBenutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#).

Der Amazon S3 S3-Service unterstützt Bucket-Richtlinien, Zugriffspunkt-Richtlinien und Zugriffsberechtigungen:

- Bucket-Richtlinien sind ressourcenbasierte Richtlinien, die an einen Amazon S3 S3-Bucket angehängt sind. Eine Bucket-Richtlinie definiert, welche Principals Aktionen für den Bucket ausführen können.
- Access Point-Richtlinien sind ressourcenbasierte Richtlinien, die in Verbindung mit der zugrunde liegenden Bucket-Richtlinie bewertet werden.
- Zugriffsberechtigungen sind ein vereinfachtes Modell zur Definition von Zugriffsberechtigungen für Daten in Amazon S3 nach Präfix, Bucket oder Objekt. Informationen zu S3-Zugriffsberechtigungen finden Sie unter [Verwalten des Zugriffs mit S3-Zugriffsberechtigungen](#).

Prinzipien für Bucket-Richtlinien

Das `Principal` Element gibt den Benutzer, das Konto, den Dienst oder eine andere Entität an, der der Zugriff auf eine Ressource entweder gewährt oder verweigert wird. Die folgenden sind Beispiele legen den `Principal` fest. Weitere Informationen finden Sie unter [Principal](#) im IAMBenutzerhandbuch.

Erteilen Sie Berechtigungen für einen AWS-Konto

Um einem Berechtigungen zu erteilen AWS-Konto, identifizieren Sie das Konto mithilfe des folgenden Formats.

```
"AWS": "account-ARN"
```

Im Folgenden sind einige Beispiele aufgeführt.

```
"Principal": {"AWS": "arn:aws:iam::AccountIDWithoutHyphens:root"}
```

```
"Principal": {"AWS":  
["arn:aws:iam::AccountID1WithoutHyphens:root", "arn:aws:iam::AccountID2WithoutHyphens:root"]}]
```

Erteilen Sie einem IAM Benutzer Berechtigungen

Um einem IAM Benutzer in Ihrem Konto Berechtigungen zu erteilen, müssen Sie ein `"AWS": "user-ARN"` Name-Wert-Paar angeben.

```
"Principal": {"AWS": "arn:aws:iam::account-number-without-hyphens:user/username"}
```

Ausführliche Beispiele mit step-by-step Anweisungen finden Sie unter [Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen](#) und [Beispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören](#)

Note

Wenn eine IAM Identität gelöscht wird, nachdem Sie Ihre Bucket-Richtlinie aktualisiert haben, zeigt die Bucket-Richtlinie anstelle einer eindeutigen Kennung im Hauptelement anARN. Diese eindeutigen Daten IDs werden niemals wiederverwendet, sodass Sie Prinzipale mit eindeutigen Kennungen problemlos aus all Ihren Richtlinienerklärungen entfernen können.

Weitere Informationen zu eindeutigen Kennungen finden Sie unter [IAM Identifikatoren](#) im Benutzerhandbuch. IAM

Erteilen anonymer Berechtigungen

Warning

Seien Sie vorsichtig, wenn Sie anonymen Zugriff auf Ihren Amazon-S3-Bucket erteilen. Wenn Sie anonymen Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihren Bucket zugreifen. Wir empfehlen dringend, nie einen anonymen Schreibzugriff auf Ihren S3-Bucket zu gewähren.

Um jedem Benutzer Berechtigung zu erteilen, auch als anonymer Zugriff bezeichnet, legen Sie den Platzhalter ("*") auf den Wert `Principal` fest. Wenn Sie beispielsweise ihren Bucket als Website konfigurieren, müssen Sie alle Objekte im Bucket öffentlich zugänglich machen.

```
"Principal": "*" 
```

```
"Principal": {"AWS": "*" }
```

Die Verwendung `"Principal": "*"` mit Allow Wirkung in einer ressourcenbasierten Richtlinie ermöglicht es jedem, auch wenn er nicht angemeldet ist AWS, um auf Ihre Ressource zuzugreifen.

Durch die Verwendung `"Principal" : { "AWS" : "*" }` mit Allow Wirkung in einer ressourcenbasierten Richtlinie können alle Root-Benutzer, IAM Benutzer, Sitzungen mit übernommener Rolle oder Verbundbenutzer in einem beliebigen Konto in derselben Partition auf Ihre Ressource zugreifen.

Für anonyme Benutzer sind diese beiden Methoden gleichwertig. Weitere Informationen finden Sie im Benutzerhandbuch unter [Alle Hauptbenutzer](#). IAM

Sie können keinen Platzhalter verwenden, um einem Teil eines Prinzipalnamens oder zuzuordnen.
ARN

⚠ Important

Weil jeder eine erstellen kann AWS-Konto, das Sicherheitsniveau dieser beiden Methoden ist gleichwertig, obwohl sie unterschiedlich funktionieren.

Ressourcenberechtigungen beschränken

Sie können die Ressourcenrichtlinie auch verwenden, um den Zugriff auf Ressourcen zu beschränken, die sonst für IAM Prinzipale verfügbar wären. Verwenden Sie eine Deny-Anweisung, um den Zugriff zu verhindern.

Das folgende Beispiel blockiert den Zugriff, wenn kein sicheres Transportprotokoll verwendet wird:

```
{"Effect": "Deny",
  "Principal": "*",
  "Action": "s3:*",
  "Resource": <bucket ARN>,
  "Condition": {
    "Boolean": { "aws:SecureTransport" : "false"}
  }
}
```

Es hat sich für diese Richtlinie bewährt, "Principal": "*" so zu verwenden, dass diese Einschränkung für alle gilt, anstatt zu versuchen, mit dieser Methode nur bestimmten Konten oder Prinzipalen den Zugriff zu verweigern.

Erfordern Sie Zugriff über CloudFront URLs

Sie können verlangen, dass Ihre Benutzer nur auf Ihre Amazon S3-Inhalte zugreifen, indem sie Amazon S3 CloudFront URLs anstelle von Amazon S3 verwenden URLs. Erstellen Sie dazu eine CloudFront Origin-Zugriffskontrolle (OAC). Ändern Sie dann die Berechtigungen für Ihre S3-Daten. In Ihrer Bucket-Richtlinie können Sie Folgendes CloudFront als Principal festlegen:

```
"Principal":{"Service":"cloudfront.amazonaws.com"}
```

Verwenden Sie ein Condition Element in der Richtlinie, um nur dann Zugriff auf den Bucket CloudFront zu gewähren, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die den S3-Ursprung enthält.

```
"Condition": {
  "StringEquals": {
    "AWS:SourceArn":
"arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
  }
}
```

Weitere Informationen zur Anforderung von S3-Zugriff über CloudFront URLs finden Sie unter [Beschränken des Zugriffs auf einen Amazon Simple Storage Service-Ursprung](#) im Amazon CloudFront Developer Guide. Weitere Informationen zu den Sicherheits- und Datenschutzvorteilen der Nutzung von Amazon CloudFront finden Sie unter [Konfiguration des sicheren Zugriffs und Beschränkung des Zugriffs auf Inhalte](#).

Beispiele für ressourcenbasierte Richtlinien für Amazon S3

- Richtlinienbeispiele für Amazon S3 S3-Buckets finden Sie unter [Bucket-Richtlinien für Amazon S3](#).
- Richtlinienbeispiele für Access Points finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#).

Politische Maßnahmen für Amazon S3

Unterstützt Richtlinienaktionen: Ja

Administratoren können verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörigen AWS API-Betrieb. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur Berechtigungen erforderlich sind und für die es keine entsprechende Operation gibt. API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Im Folgenden werden verschiedene Arten von Zuordnungsbeziehungen zwischen API S3-Vorgängen und den erforderlichen Richtlinienaktionen dargestellt.

- ne-to-one O-Zuordnung mit demselben Namen. Um beispielsweise den `PutBucketPolicy` API Vorgang verwenden zu können, ist die `s3:PutBucketPolicy` Richtlinienaktion erforderlich.
- ne-to-one O-Mapping mit unterschiedlichen Namen. Um beispielsweise den `ListObjectsV2` API Vorgang verwenden zu können, ist die `s3:ListBucket` Richtlinienaktion erforderlich.
- Keine ne-to-many Zuordnung. Um beispielsweise die `HeadObject` API Operation verwenden zu können, `s3:GetObject` ist erforderlich. Wenn Sie S3 Object Lock verwenden und den Status oder die Aufbewahrungseinstellungen eines Objekts abrufen möchten, sind außerdem die entsprechenden Aktionen `s3:GetObjectLegalHold` oder die `s3:GetObjectRetention` Richtlinienaktionen erforderlich, bevor Sie den `HeadObject` API Vorgang verwenden können.
- any-to-one M-Zuordnung. Um beispielsweise die `HeadBucket` API Operationen `ListObjectsV2` oder zu verwenden, ist die `s3:ListBucket` Richtlinienaktion erforderlich.

Eine Liste der Amazon S3-Aktionen zur Verwendung in Richtlinien finden Sie unter [Von Amazon S3 definierte Aktionen](#) in der Service Authorization Reference. Eine vollständige Liste der Amazon S3 API S3-Operationen finden Sie unter [Amazon S3 API S3-Aktionen](#) in der Amazon Simple Storage API Service-Referenz.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Richtlinienaktionen in Amazon S3 verwenden das folgende Präfix vor der Aktion:

```
s3
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "s3:action1",  
  "s3:action2"  
]
```

Bucket-Operationen

Bucket-Operationen sind API S3-Operationen, die auf dem Bucket-Ressourcentyp ausgeführt werden. Beispiel: `CreateBucket`, `ListObjectsV2` und `PutBucketPolicy`. S3-Richtlinienaktionen für Bucket-Operationen erfordern, dass das Resource Element in Bucket-Richtlinien oder IAM identitätsbasierten Richtlinien der Amazon Resource Name (ARN) -Bezeichner vom Typ S3-Bucket im folgenden Beispielformat ist.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
```

Die folgende Bucket-Richtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:ListBucket` Erlaubnis, Folgendes durchzuführen [ListObjectsV2](#) API-Operation und Listet Objekte in einem S3-Bucket auf.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow Akua to list objects in the bucket",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::12345678901:user/Akua"
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  ]
}
```

Bucket-Operationen in Access Point-Richtlinien

In einer Zugriffspunktrichtlinie gewährte Berechtigungen sind nur wirksam, wenn der zugrunde liegende Bucket dieselben Berechtigungen zulässt. Wenn Sie S3 Access Points verwenden, müssen Sie die Zugriffskontrolle vom Bucket an den Access Point delegieren oder dieselben Berechtigungen in den Access Point-Richtlinien zur Richtlinie des zugrunde liegenden Buckets hinzufügen. Weitere Informationen finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#). In Zugriffspunktrichtlinien erfordern S3-Richtlinienaktionen für Bucket-Operationen, dass Sie den Zugriffspunkt ARN für das Resource Element im folgenden Format verwenden.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point"
```

Die folgende Zugriffspunktrichtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:ListBucket` Erlaubnis, Folgendes auszuführen [ListObjectsV2](#) API Vorgang über den genannten S3-Zugangspunkt *example-access-point*. Diese Berechtigung ermöglicht es *Akua*, die Objekte in dem Bucket aufzulisten, dem sie zugeordnet sind *example-access-point*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow Akua to list objects in the bucket through access point",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::12345678901:user/Akua"
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-
point"
    }
  ]
}
```

Note

Nicht alle Bucket-Operationen werden von S3 Access Point unterstützt. Weitere Informationen finden Sie unter [Zugriffspunkt-Kompatibilität mit S3-Vorgänge](#).

Objektoperationen

Objektoperationen sind API S3-Operationen, die auf den Objektressourcentyp einwirken. Beispiel: `GetObject`, `PutObject` und `DeleteObject`. Für S3-Richtlinienaktionen für Objektoperationen muss das `Resource` Element in den Richtlinien das S3-Objekt ARN in den folgenden Beispielformaten sein.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
```

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/*"
```

Note

Das Objekt ARN muss nach dem Bucket-Namen einen Schrägstrich enthalten, wie in den vorherigen Beispielen gezeigt.

Die folgende Bucket-Richtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:PutObject` entsprechende Berechtigung. Diese Erlaubnis ermöglicht *Akua* die Verwendung von [PutObject](#) API Vorgang zum Hochladen von Objekten in den genannten S3-Bucket *amzn-s3-demo-bucket*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow Akua to upload objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::12345678901:user/Akua"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

Objektoperationen in Zugriffspunktrichtlinien

Wenn Sie S3 Access Points verwenden, um den Zugriff auf Objektoperationen zu steuern, können Sie Zugriffspunktrichtlinien verwenden. Wenn Sie Zugriffspunktrichtlinien verwenden, müssen Sie bei S3-Richtlinienaktionen für Objektoperationen den Zugriffspunkt ARN für das Resource Element im folgenden Format verwenden: `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource`. Bei Objektoperationen, die Zugriffspunkte verwenden, müssen Sie den `/object/` Wert hinter dem gesamten Zugriffspunkt ARN in das Resource Element aufnehmen. Hier sind einige Beispiele.


```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point/object/*"
```

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point/object/prefix/*"
```

Die folgende Zugriffspunktrichtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:GetObject` entsprechende Berechtigung. Diese Berechtigung ermöglicht *Akua* die Ausführung von [GetObject](#) API Vorgang über den Access Point, der für alle Objekte in dem Bucket benannt *example-access-point* ist, der dem Access Point zugeordnet ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow Akua to get objects through access point",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::12345678901:user/Akua"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point/object/*"
    }
  ]
}
```

Note

Nicht alle Objektoperationen werden von Access Points unterstützt. Weitere Informationen finden Sie unter [Zugriffspunkt-Kompatibilität mit S3-Vorgänge](#).

Operationen an Zugriffspunkten

Access Point-Operationen sind API S3-Operationen, die auf dem `accesspoint` Ressourcentyp ausgeführt werden. Beispiel: `CreateAccessPoint`, `DeleteAccessPoint` und `GetAccessPointPolicy`. S3-Richtlinienaktionen für Access Point-Operationen können

nur in IAM identitätsbasierten Richtlinien verwendet werden, nicht in Bucket-Richtlinien oder Zugriffspunktrichtlinien. Für Access Point-Operationen muss das Resource Element der Access Point ARN im folgenden Beispielformat sein.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point"
```

Die folgende IAM identitätsbasierte Richtlinie gewährt die `s3:GetAccessPointPolicy` Erlaubnis zur Durchführung von [GetAccessPointPolicy](#) API-Vorgang auf dem genannten S3-Zugriffspunkt. *example-access-point*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Grant permission to retrieve the access point policy of access
point example-access-point",
      "Effect": "Allow",
      "Action": [
        "s3:GetAccessPointPolicy"
      ],
      "Resource": "arn:aws:s3:*:123456789012:accesspoint/example-access-point"
    }
  ]
}
```

Wenn Sie Access Points verwenden, um den Zugriff auf Bucket-Operationen zu steuern, siehe [Bucket-Operationen in Access Point-Richtlinien](#); Informationen zur Steuerung des Zugriffs auf Objektoperationen finden Sie unter [Objektoperationen in Zugriffspunktrichtlinien](#). Weitere Informationen zur Konfiguration von Zugriffspunktrichtlinien finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#).

Operationen des Object Lambda Access Points

Mit S3 Object Lambda können Sie GET-, LIST- und HEAD-Anforderungen von Amazon S3 eigenen Code hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Sie können Anfragen über einen Object Lambda Access Point stellen, was genauso funktioniert wie Anfragen über andere Access Points. Weitere Informationen finden Sie unter [Transformieren von Objekten mit S3 Object Lambda](#).

Weitere Informationen zur Konfiguration von Richtlinien für Object Lambda Access Point-Operationen finden Sie unter [Konfiguration von IAM Richtlinien für Object Lambda Access Points](#).

Access Point-Operationen in mehreren Regionen

Ein Access Point mit mehreren Regionen bietet einen globalen Endpunkt, über den Anwendungen Anfragen von S3-Buckets bearbeiten können, die sich in mehreren AWS-Regionen befinden. Sie können einen Multi-Region-Access Point verwenden, um multiregionale Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Weitere Informationen finden Sie unter [Multi-Regions-Zugriffspunkte in Amazon S3](#).

Weitere Informationen zur Konfiguration von Richtlinien für den Betrieb von Access Points in mehreren Regionen finden Sie unter [Beispielrichtlinien für Multi-Region Access Points](#)

Batch-Job-Operationen

Auftragsoperationen (Batch Operations) sind API S3-Operationen, die auf dem Ressourcentyp Job ausgeführt werden. Beispiel: `DescribeJob` und `CreateJob`. S3-Richtlinienaktionen für Auftragsoperationen können nur in IAM identitätsbasierten Richtlinien verwendet werden, nicht in Bucket-Richtlinien. Für Auftragsoperationen muss das Resource Element in IAM identitätsbasierten Richtlinien außerdem das job ARN folgende Beispielformat haben.

```
"Resource": "arn:aws:s3:*:123456789012:job/*"
```

Die folgende IAM identitätsbasierte Richtlinie gewährt die `s3:DescribeJob` Erlaubnis, den [DescribeJob](#) API-Vorgang für den genannten Auftrag S3 Batch Operations auszuführen. *example-job*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow describing the Batch operation job example-job",
      "Effect": "Allow",
      "Action": [
        "s3:DescribeJob"
      ],
      "Resource": "arn:aws:s3:*:123456789012:job/example-job"
    }
  ]
}
```

Konfigurationsvorgänge für S3 Storage Lens

Weitere Informationen zur Konfiguration von S3 Storage Lens-Konfigurationsvorgängen finden Sie unter [Berechtigungen für Amazon S3 Storage Lens](#).

Kontovorgänge

Kontovorgänge sind API S3-Operationen, die auf Kontoebene ausgeführt werden. Zum Beispiel `GetPublicAccessBlock` (für ein Konto). Konto ist kein von Amazon S3 definierter Ressourcentyp. S3-Richtlinienaktionen für Kontooperationen können nur in IAM identitätsbasierten Richtlinien verwendet werden, nicht in Bucket-Richtlinien. Außerdem ist für Kontooperationen das `Resource` Element in IAM identitätsbasierten Richtlinien erforderlich. "*"

Die folgende IAM identitätsbasierte Richtlinie erteilt die `s3:GetAccountPublicAccessBlock` Erlaubnis, die Ausführung auf Kontoebene durchzuführen [GetPublicAccessBlock](#) API-Betrieb und Abrufen der Einstellungen für die Sperrung des öffentlichen Zugriffs auf Kontoebene.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"Allow retrieving the account-level Public Access Block settings",
      "Effect":"Allow",
      "Action":[
        "s3:GetAccountPublicAccessBlock"
      ],
      "Resource":[
        "*"
      ]
    }
  ]
}
```

Richtlinienbeispiele für Amazon S3

- Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#)
- Beispiele für ressourcenbasierte Amazon S3 S3-Richtlinien finden Sie unter [Bucket-Richtlinien für Amazon S3](#) und [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#)

Richtlinienressourcen für Amazon S3

Unterstützt Richtlinienressourcen: Ja

Administratoren können Folgendes verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem [Amazon-Ressourcennamen \(ARN\)](#) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Einige Amazon S3 API S3-Aktionen unterstützen mehrere Ressourcen. `s3:GetObject` greift beispielsweise auf *example-resource-1* und zu *example-resource-2*, sodass ein Principal über Berechtigungen für den Zugriff auf beide Ressourcen verfügen muss. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie ARNs durch Kommas, wie im folgenden Beispiel gezeigt.

```
"Resource": [  
  "example-resource-1",  
  "example-resource-2"
```

Ressourcen in Amazon S3 sind Buckets, Objekte, Zugriffspunkte oder Jobs. Verwenden Sie in einer Richtlinie den Amazon-Ressourcennamen (ARN) des Buckets, Objekts, Access Points oder Jobs, um die Ressource zu identifizieren.

Eine vollständige Liste der Amazon S3-Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon S3 definierte Ressourcen](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter [Von Amazon S3 definierte Aktionen](#).

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Platzhalterzeichen in der Ressource ARNs

Sie können Platzhalterzeichen als Teil der Ressource verwenden. ARN Sie können die Platzhalterzeichen (*und?) in jedem ARN Segment (die durch Doppelpunkte getrennten Teile) verwenden. Ein Sternchen (*) steht für kein Zeichen oder eine beliebige Kombination von mehreren Zeichen und ein Fragezeichen (?) entspricht einem beliebigen einzelnen Zeichen. Sie können in jedem Segment mehrere ? Zeichen * oder verwenden. Ein Platzhalterzeichen kann sich jedoch nicht über mehrere Segmente erstrecken.

- Im Folgenden ARN wird das * Platzhalterzeichen im relative-ID Teil von verwendet, ARN um alle Objekte im Bucket zu identifizieren. *amzn-s3-demo-bucket*

```
arn:aws:s3:::amzn-s3-demo-bucket/*
```

- Im Folgenden werden ARN alle * S3-Buckets und -Objekte bezeichnet.

```
arn:aws:s3:::*
```

- Im Folgenden ARN werden beide Platzhalterzeichen * und?, im relative-ID Teil verwendet. Dadurch ARN werden alle Objekte in Buckets identifiziert, z. B. *amzn-s3-demo-example1bucket*,, und so weiter. *amzn-s3-demo-example2bucket amzn-s3-demo-example3bucket*

```
arn:aws:s3:::amzn-s3-demo-example?bucket/*
```

Richtlinienvariablen für Ressourcen ARNs

Sie können Richtlinienvariablen in Amazon S3 verwenden ARNs. Bei der Überprüfung der Richtlinien werden diese vordefinierten Variablen durch ihre entsprechenden Werte ersetzt. Angenommen, Sie organisieren Ihren Bucket als eine Sammlung von Ordnern mit einem Ordner für jeden Ihrer Benutzer. Der Ordnername ist derselbe wie der Benutzername. Um Benutzern Berechtigungen für ihre Ordner zu gewähren, können Sie eine Richtlinienvariable in der Ressource angeben ARN:

```
arn:aws:s3:::bucket_name/developers/${aws:username}/
```

Zur Laufzeit, wenn die Richtlinie ausgewertet wird, ARN wird die Variable `${aws:username}` in der Ressource durch den Benutzernamen der Person ersetzt, die die Anfrage stellt.

Richtlinienbeispiele für Amazon S3

- Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#)
- Beispiele für ressourcenbasierte Amazon S3 S3-Richtlinien finden Sie unter [Bucket-Richtlinien für Amazon S3](#) und [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#)

Schlüssel für Richtlinienbedingungen für Amazon S3

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können Folgendes verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition` Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition` Element angeben, AWS wertet sie mithilfe einer logischen AND Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Sie können einem IAM Benutzer beispielsweise nur dann Zugriff auf eine Ressource gewähren, wenn sie mit seinem IAM Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinienelemente: Variablen und Tags](#).

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Um alle zu sehen AWS globale Bedingungsschlüssel finden Sie unter [AWS Kontexttasten für globale Bedingungen](#) im IAMBenutzerhandbuch.

Jeder Amazon S3 S3-Bedingungsschlüssel ist demselben Anforderungsheader mit demselben Namen zugeordnet, der von API dem zugelassen wird, für den die Bedingung gesetzt werden kann. Amazon-S3-spezifische Bedingungsschlüssel bestimmen das Verhalten der Anforderungs-Header mit demselben Namen. Beispielsweise definiert der Bedingungsschlüssel, der `s3:VersionId` verwendet wird, um die bedingte Berechtigung zu

`s3:GetObjectVersion`

erteilen, das Verhalten des `versionId` Abfrageparameters, den Sie in einer GET Objektanforderung festlegen.

Eine Liste der Amazon S3-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon S3 definierte Aktionen](#).

Beispiel: Beschränkung von Objekt-Uploads auf Objekte mit einer bestimmten Speicherklasse

Angenommen, Konto A, dargestellt durch die Konto-ID `123456789012`, besitzt einen Bucket. Das Konto, das der Administrator einschränken möchte *Dave*, ein Benutzer in Konto A, so dass *Dave* kann Objekte nur dann in den Bucket hochladen, wenn das Objekt in der STANDARD_IA Speicherklasse gespeichert ist. Um das Hochladen von Objekten auf eine bestimmte Speicherklasse einzuschränken, kann der Administrator von Konto A den Bedingungsschlüssel `s3:x-amz-storage-class` verwenden wie in der folgenden Bucket-Beispielrichtlinie gezeigt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-storage-class": [
            "STANDARD_IA"
          ]
        }
      }
    }
  ]
}
```



```
    }  
  ]  
}
```

Im Beispiel gibt der Block `Condition` die Bedingung `StringEquals` an, die auf das angegebene Schlüssel-Wert-Paar angewendet wird, `"s3:x-amz-acl":["public-read"]`. Es gibt einen Satz vordefinierter Schlüssel, die Sie zum Ausdruck einer Bedingung verwenden können. Das Beispiel verwendet den Bedingungsschlüssel `s3:x-amz-acl`. Diese Bedingung erfordert, dass der Benutzer den `x-amz-acl` Header mit dem Wert `public-read` in jede `PutObject` Anfrage einfügt.

Richtlinienbeispiele für Amazon S3

- Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#)
- Beispiele für ressourcenbasierte Amazon S3 S3-Richtlinien finden Sie unter [Bucket-Richtlinien für Amazon S3](#) und [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#)

ACLs auf Amazon S3

Unterstützt ACLs: Ja

In Amazon S3 steuern Zugriffskontrolllisten (ACLs), welche AWS-Konten über Berechtigungen für den Zugriff auf eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Important

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs.

Informationen zur Verwendung ACLs zur Zugriffskontrolle in Amazon S3 finden Sie unter [Verwaltung des Zugriffs mit ACLs](#).

ABAC mit Amazon S3

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen auf der Grundlage von Attributen definiert werden. In AWS, diese Attribute werden Tags genannt.

Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele Entitäten anhängen AWS Ressourcen schützen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABAC ist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu finden Sie ABAC unter [Was ist? ABAC](#) im IAM Benutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter [Verwenden der attributbasierten Zugriffskontrolle \(ABAC\)](#). IAM

Beispiele für identitätsbasierte Richtlinien zur Beschränkung des Zugriffs auf S3 Batch Operations-Jobs auf der Grundlage von Tags finden Sie unter [Steuern von Berechtigungen für S3-Batch-Vorgänge mithilfe von Auftrags-Markierungen](#)

ABAC und Objekt-Tags

In ABAC Richtlinien verwenden Objekte `s3:Tags` anstelle von `aws:Tags`. Um den Zugriff auf Objekte auf der Grundlage von Objekt-Tags zu steuern, geben Sie Tag-Informationen in der [Condition Element](#) einer Richtlinie unter Verwendung der folgenden Tags:

- `s3:ExistingObjectTag/tag-key`
- `s3:s3:RequestObjectTagKeys`
- `s3:RequestObjectTag/tag-key`

Informationen zur Verwendung von Objekt-Tags zur Zugriffskontrolle, einschließlich beispielhafter Berechtigungsrichtlinien, finden Sie unter [Markierungs- und Zugriffskontrollrichtlinien](#).

Temporäre Anmeldeinformationen mit Amazon S3 verwenden

Unterstützt temporäre Anmeldeinformationen: Ja

Einige AWS-Services funktionieren nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Für zusätzliche Informationen, einschließlich AWS-Services mit temporären Anmeldeinformationen arbeiten, finden Sie unter [AWS-Services mit denen IAM](#) im IAMBenutzerhandbuch gearbeitet werden kann.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich bei der AWS Management Console mit einer beliebigen Methode außer einem Benutzernamen und einem Passwort. Zum Beispiel, wenn Sie darauf zugreifen AWS Wenn Sie den Single Sign-On-Link (SSO) Ihres Unternehmens verwenden, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAMBenutzerhandbuch.

Sie können temporäre Anmeldeinformationen manuell erstellen, indem Sie den AWS CLI or AWS API. Sie können dann diese temporären Anmeldeinformationen für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen unter IAM](#).

Zugriffssitzungen für Amazon S3 weiterleiten

Unterstützt Forward-Access-Sitzungen (FAS): Ja

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen in AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Rechte des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage AWS-Service um Anfragen an nachgelagerte Dienste zu stellen. FASANfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, die Interaktionen mit anderen erfordert AWS-Services oder zu vervollständigende Ressourcen. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

- FASwird von Amazon S3 verwendet, um Anrufe zu tätigen AWS KMS um ein Objekt zu entschlüsseln, wenn SSE - verwendet KMS wurde, um es zu verschlüsseln. Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).
- S3 Access Grants verwendet auchFAS. Nachdem Sie eine Zugriffsberechtigung für Ihre S3-Daten für eine bestimmte Identität erstellt haben, fordert der Empfänger temporäre Anmeldeinformationen von S3 Access Grants an. S3 Access Grants erhält temporäre Anmeldeinformationen für den

Antragsteller von AWS STS und sendet die Anmeldeinformationen an den Anforderer. Weitere Informationen finden Sie unter [Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants](#).

Servicerollen für Amazon S3

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAMRolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an ein AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Durch das Ändern der Berechtigungen für eine Servicerolle kann die Amazon S3 S3-Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn Amazon S3 Sie dazu anleitet.

Servicebezogene Rollen für Amazon S3

Unterstützt serviceverknüpfte Rollen: Teilweise

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen durchzuführen. Mit Diensten verknüpfte Rollen erscheinen in Ihrem AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.

Amazon S3 unterstützt serviceverknüpfte Rollen für Amazon S3 Storage Lens. Einzelheiten zum Erstellen oder Verwalten von serviceverknüpften Amazon S3 S3-Rollen finden Sie unter [Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens](#).

Amazon S3 Service als Principal

Name des Dienstes in der Richtlinie	S3-Funktion	Weitere Informationen
s3.amazonaws.com	S3-Replikation	Live-Replikation einrichten

Name des Dienstes in der Richtlinie	S3-Funktion	Weitere Informationen
s3.amazonaws.com	S3-Ereignisbenachrichtigungen	Amazon-S3-Ereignis-Benachrichtigungen
s3.amazonaws.com	S3-Bestand	Amazon S3 Inventory
access-grants.s3.amazonaws.com	S3 Access Grants	Registrieren eines Speichers
batchoperations.s3.amazonaws.com	S3-Batchvorgänge	Erteilen von Berechtigungen für Amazon-S3-Batchvorgänge
logging.s3.amazonaws.com	Protokollierung des S3-Serverzugriffs	Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung
storage-lens.s3.amazonaws.com	S3 Storage Lens	Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport

Erforderliche Berechtigungen für Amazon S3 API S3-Operationen

Note

Auf dieser Seite geht es um Amazon S3 S3-Richtlinienaktionen für Allzweck-Buckets. Weitere Informationen zu Amazon S3 S3-Richtlinienaktionen für Directory-Buckets finden Sie unter [Aktionen für S3 Express One Zone](#).

Um einen API S3-Vorgang ausführen zu können, benötigen Sie die richtigen Berechtigungen. Auf dieser Seite werden API S3-Operationen den erforderlichen Berechtigungen zugeordnet. Um Berechtigungen zur Ausführung eines API S3-Vorgangs zu erteilen, müssen Sie eine gültige Richtlinie (z. B. eine S3-Bucket-Richtlinie oder eine IAM identitätsbasierte Richtlinie) verfassen und die entsprechenden Aktionen im Action Element der Richtlinie angeben. Diese Aktionen werden als Richtlinienaktionen bezeichnet. Nicht jeder API S3-Vorgang wird durch eine einzige Berechtigung

(eine einzelne Richtlinienaktion) repräsentiert, und einige Berechtigungen (einige Richtlinienaktionen) sind für viele verschiedene API Operationen erforderlich.

Wenn Sie Richtlinien erstellen, müssen Sie das Resource Element auf der Grundlage des richtigen Ressourcentyps angeben, der für die entsprechenden Amazon S3 S3-Richtlinienaktionen erforderlich ist. Auf dieser Seite werden die Berechtigungen für API S3-Operationen nach Ressourcentypen kategorisiert. Weitere Informationen zu den Ressourcentypen finden Sie unter [Von Amazon S3 definierte Ressourcentypen](#) in der Service Authorization Reference. Eine vollständige Liste der Amazon S3-Richtlinienaktionen, Ressourcen und Bedingungsschlüssel zur Verwendung in Richtlinien finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference. Eine vollständige Liste der Amazon S3 API S3-Operationen finden Sie unter [Amazon S3 API S3-Aktionen](#) in der Amazon Simple Storage API Service-Referenz.

Themen

- [Bucket-Operationen und -Berechtigungen](#)
- [Objektoperationen und Berechtigungen](#)
- [Operationen und Berechtigungen des Access Points](#)
- [Operationen und Berechtigungen für Object Lambda Access Point](#)
- [Operationen und Berechtigungen für Access Points in mehreren Regionen](#)
- [Batch-Job-Operationen und Berechtigungen](#)
- [Konfiguration, Vorgänge und Berechtigungen von S3 Storage Lens](#)
- [S3 Storage Lens gruppiert Operationen und Berechtigungen](#)
- [Kontooperationen und Berechtigungen](#)

Bucket-Operationen und -Berechtigungen

Bucket-Operationen sind API S3-Operationen, die auf dem Bucket-Ressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Bucket-Operationen in Bucket-Richtlinien oder IAM identitätsbasierten Richtlinien angeben.

In den Richtlinien muss das Resource Element der Bucket Amazon Resource Name (ARN) sein. Weitere Informationen zum Resource Elementformat und zu Beispielrichtlinien finden Sie unter [Bucket-Operationen](#).

Note

Beachten Sie Folgendes, um Berechtigungen für Bucket-Operationen in Zugriffspunktrichtlinien zu erteilen:

- Berechtigungen, die in einer Zugriffspunktrichtlinie für Bucket-Operationen gewährt werden, sind nur wirksam, wenn der zugrunde liegende Bucket dieselben Berechtigungen zulässt. Wenn Sie einen Access Point verwenden, müssen Sie die Zugriffskontrolle vom Bucket an den Access Point delegieren oder die gleichen Berechtigungen in der Zugriffspunktrichtlinie zur Richtlinie des zugrunde liegenden Buckets hinzufügen.
- In Zugriffspunktrichtlinien, die Berechtigungen für Bucket-Operationen gewähren, muss das Resource Element das `accesspoint` ARN sein. Weitere Informationen zum Resource Elementformat und zu Beispielrichtlinien finden Sie unter [Bucket-Operationen in Access Point-Richtlinien](#). Weitere Informationen zu Zugriffspunktrichtlinien finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#).
- Nicht alle Bucket-Operationen werden von Access Points unterstützt. Weitere Informationen finden Sie unter [Zugriffspunkt-Kompatibilität mit S3-Vorgänge](#).

Im Folgenden finden Sie die Zuordnung von Bucket-Vorgängen und erforderlichen Richtlinienaktionen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateBucket	(Erforderlich) <code>s3:CreateBucket</code>	Erforderlich, um einen neuen S3-Bucket zu erstellen.
	(Bedingt erforderlich) <code>s3:PutBucketAcl</code>	Erforderlich, wenn Sie die Zugriffskontrollliste (ACL) verwenden möchten, um bei einer <code>CreateBucket</code> Anfrage Berechtigungen für einen Bucket anzugeben.
	(Bedingt erforderlich) <code>s3:PutBucketObject</code>	Erforderlich, wenn Sie beim Erstellen eines Buckets

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	LockConfiguration , s3:PutBucketVersioning	die Objektsperre aktivieren möchten.
	(Bedingt erforderlich) s3:PutBucketOwnershipControls	Erforderlich, wenn Sie beim Erstellen eines Buckets S3-Objekteigentümerschaft angeben möchten.
DeleteBucket	(Erforderlich) s3:DeleteBucket	Erforderlich, um einen S3-Bucket zu löschen.
DeleteBucketAnalyticsConfiguration	(Erforderlich) s3:PutAnalyticsConfiguration	Erforderlich, um eine S3-Analytics-Konfiguration aus einem S3-Bucket zu löschen.
DeleteBucketCors	(Erforderlich) s3:PutBucketCors	Erforderlich, um die ursprungsübergreifende Resource Sharing (CORS) -Konfiguration für einen Bucket zu löschen.
DeleteBucketEncryption	(Erforderlich) s3:PutEncryptionConfiguration	Erforderlich, um die Standardverschlüsselungskonfiguration für einen S3-Bucket als serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) zurückzusetzen.
DeleteBucketIntelligentTieringConfiguration	(Erforderlich) s3:PutIntelligentTieringConfiguration	Erforderlich, um die bestehende S3 Intelligent-Tiering-Konfiguration aus einem S3-Bucket zu löschen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteBucketInventoryConfiguration	(Erforderlich) <code>s3:PutInventoryConfiguration</code>	Erforderlich, um eine S3-Inventarkonfiguration aus einem S3-Bucket zu löschen.
DeleteBucketLifecycle	(Erforderlich) <code>s3:PutLifecycleConfiguration</code>	Erforderlich, um die S3-Lifecycle-Konfiguration für einen S3-Bucket zu löschen.
DeleteBucketMetricsConfiguration	(Erforderlich) <code>s3:PutMetricsConfiguration</code>	Erforderlich, um eine Metrikkonfiguration für die CloudWatch Amazon-Anforderungsmetriken aus einem S3-Bucket zu löschen.
DeleteBucketOwnershipControls	(Erforderlich) <code>s3:PutBucketOwnershipControls</code>	Erforderlich, um die Einstellung „Objektbesitz“ für einen S3-Bucket zu entfernen. Nach dem Entfernen wird die Einstellung Object Ownership zu <code>ObjectWriter</code> .
DeleteBucketPolicy	(Erforderlich) <code>s3:DeleteBucketPolicy</code>	Erforderlich, um die Richtlinie eines S3-Buckets zu löschen.
DeleteBucketReplication	(Erforderlich) <code>s3:PutReplicationConfiguration</code>	Erforderlich, um die Replikationskonfiguration eines S3-Buckets zu löschen.
DeleteBucketTagging	(Erforderlich) <code>s3:PutBucketTagging</code>	Erforderlich, um Tags aus einem S3-Bucket zu löschen.
DeleteBucketWebsite	(Erforderlich) <code>s3:DeleteBucketWebsite</code>	Erforderlich, um die Website-Konfiguration für einen S3-Bucket zu entfernen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeletePublicAccessBlock (Bucket-Ebene)	(Erforderlich) <code>s3:PutBucketPublicAccessBlock</code>	Erforderlich, um die Konfiguration zum Blockieren des öffentlichen Zugriffs für einen S3-Bucket zu entfernen.
GetBucketAccelerateConfiguration	(Erforderlich) <code>s3:GetAccelerateConfiguration</code>	Erforderlich, um die Accelerate-Unterressource zu verwenden, um den Amazon S3 Transfer Acceleration Acceleration-Status eines Buckets zurückzugeben, der entweder aktiviert oder ausgesetzt ist.
GetBucketAcl	(Erforderlich) <code>s3:GetBucketAcl</code>	Erforderlich, um die Zugriffskontrollliste (ACL) eines S3-Buckets zurückzugeben.
GetBucketAnalyticsConfiguration	(Erforderlich) <code>s3:GetAnalyticsConfiguration</code>	Erforderlich, um eine Analytics-Konfiguration zurückzugeben, die durch die Analytics-Konfigurations-ID aus einem S3-Bucket identifiziert wurde.
GetBucketCors	(Erforderlich) <code>s3:GetBucketCORS</code>	Erforderlich, um die ursprungsübergreifende Resource Sharing (CORS) -Konfiguration für einen S3-Bucket zurückzugeben.
GetBucketEncryption	(Erforderlich) <code>s3:GetEncryptionConfiguration</code>	Erforderlich, um die Standardverschlüsselungskonfiguration für einen S3-Bucket zurückzugeben.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetBucketIntelligentTieringConfiguration	(Erforderlich) <code>s3:GetIntelligentTieringConfiguration</code>	Erforderlich, um die S3 Intelligent-Tiering-Konfiguration eines S3-Buckets abzurufen.
GetBucketInventoryConfiguration	(Erforderlich) <code>s3:GetInventoryConfiguration</code>	Erforderlich, um eine Inventarkonfiguration zurückzugeben, die anhand der Inventarkonfigurations-ID aus dem Bucket identifiziert wurde.
GetBucketLifecycle	(Erforderlich) <code>s3:GetLifecycleConfiguration</code>	Erforderlich, um die S3-Lifecycle-Konfiguration des Buckets zurückzugeben.
GetBucketLocation	(Erforderlich) <code>s3:GetBucketLocation</code>	Erforderlich für die Rückgabe der AWS-Region in dem sich ein S3-Bucket befindet.
GetBucketLogging	(Erforderlich) <code>s3:GetBucketLogging</code>	Erforderlich, um den Protokollierungsstatus eines S3-Buckets und die Berechtigungen zurückzugeben, über die Benutzer verfügen, um diesen Status anzuzeigen und zu ändern.
GetBucketMetricsConfiguration	(Erforderlich) <code>s3:GetMetricsConfiguration</code>	Erforderlich, um eine Metrikkonfiguration abzurufen, die durch die Metrik-Konfigurations-ID aus dem Bucket angegeben ist.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetBucketNotificationConfiguration	(Erforderlich) <code>s3:GetBucketNotification</code>	Erforderlich, um die Benachrichtigungskonfiguration eines S3-Buckets zurückzugeben.
GetBucketOwnershipControls	(Erforderlich) <code>s3:GetBucketOwnershipControls</code>	Erforderlich, um die Einstellung „Objektbesitz“ für einen S3-Bucket abzurufen.
GetBucketPolicy	(Erforderlich) <code>s3:GetBucketPolicy</code>	Erforderlich, um die Richtlinie eines S3-Buckets zurückzugeben.
GetBucketPolicyStatus	(Erforderlich) <code>s3:GetBucketPolicyStatus</code>	Erforderlich, um den Richtlinienstatus für einen S3-Bucket abzurufen, der angibt, ob der Bucket öffentlich ist.
GetBucketReplication	(Erforderlich) <code>s3:GetReplicationConfiguration</code>	Erforderlich, um die Replikationskonfiguration eines S3-Buckets zurückzugeben.
GetBucketRequestPayment	(Erforderlich) <code>s3:GetBucketRequestPayment</code>	Erforderlich, um die Konfiguration der Zahlungsanfrage für einen S3-Bucket zurückzugeben.
GetBucketVersioning	(Erforderlich) <code>s3:GetBucketVersioning</code>	Erforderlich, um den Versionsstatus eines S3-Buckets zurückzugeben.
GetBucketTagging	(Erforderlich) <code>s3:GetBucketTagging</code>	Erforderlich, um den Tagsatz zurückzugeben, der einem S3-Bucket zugeordnet ist.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetBucketWebsite	(Erforderlich) <code>s3:GetBucketWebsite</code>	Erforderlich, um die Website-Konfiguration für einen S3-Bucket zurückzugeben.
GetObjectLockConfiguration	(Erforderlich) <code>s3:GetObjectLockConfiguration</code>	Erforderlich, um die Object Lock-Konfiguration für einen S3-Bucket abzurufen.
GetPublicAccessBlock(Bucket-Ebene)	(Erforderlich) <code>s3:GetBucketPublicAccessBlock</code>	Erforderlich, um die Konfiguration für den öffentlichen Blockzugriff für einen S3-Bucket abzurufen.
HeadBucket	(Erforderlich) <code>s3:ListBucket</code>	Erforderlich, um festzustellen, ob ein Bucket vorhanden ist und ob Sie berechtigt sind, darauf zuzugreifen.
ListBucketAnalyticsConfigurations	(Erforderlich) <code>s3:GetAnalyticsConfiguration</code>	Erforderlich, um die Analytics-Konfigurationen für einen S3-Bucket aufzulisten.
ListBucketIntelligentTieringConfigurations	(Erforderlich) <code>s3:GetIntelligentTieringConfiguration</code>	Erforderlich, um die S3 Intelligent-Tiering-Konfigurationen eines S3-Buckets aufzulisten.
ListBucketInventoryConfigurations	(Erforderlich) <code>s3:GetInventoryConfiguration</code>	Erforderlich, um eine Liste der Inventarkonfigurationen für einen S3-Bucket zurückzugeben.
ListBucketMetricsConfigurations	(Erforderlich) <code>s3:GetMetricsConfiguration</code>	Erforderlich, um die Metrikkonfigurationen für einen S3-Bucket aufzulisten.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
ListObjects	(Erforderlich) <code>s3:ListBucket</code>	Erforderlich, um einige oder alle (bis zu 1.000) Objekte in einem S3-Bucket aufzulisten.
	(Bedingt erforderlich) <code>s3:GetObjectAcl</code>	Erforderlich, wenn Sie Informationen zum Objekteigentümer anzeigen möchten.
ListObjectsV2	(Erforderlich) <code>s3:ListBucket</code>	Erforderlich, um einige oder alle (bis zu 1.000) Objekte in einem S3-Bucket aufzulisten.
	(Bedingt erforderlich) <code>s3:GetObjectAcl</code>	Erforderlich, wenn Sie Informationen zum Objekteigentümer anzeigen möchten.
ListObjectVersions	(Erforderlich) <code>s3:ListBucketVersions</code>	Erforderlich, um Metadaten zu allen Versionen von Objekten in einem S3-Bucket abzurufen.
PutBucketAccelerateConfiguration	(Erforderlich) <code>s3:PutAccelerateConfiguration</code>	Erforderlich, um die Accelerate-Konfiguration eines vorhandenen Buckets festzulegen.
PutBucketAcl	(Erforderlich) <code>s3:PutBucketAcl</code>	Erforderlich, um mithilfe von Zugriffskontrolllisten (ACLs) die Berechtigungen für einen vorhandenen Bucket festzulegen.
PutBucketAnalyticsConfiguration	(Erforderlich) <code>s3:PutAnalyticsConfiguration</code>	Erforderlich, um eine Analytics-Konfiguration für einen S3-Bucket festzulegen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutBucketCors	(Erforderlich) <code>s3:PutBucketCORS</code>	Erforderlich, um die Cross-Origin Resource Sharing (CORS)-Konfiguration für einen S3-Bucket festzulegen.
PutBucketEncryption	(Erforderlich) <code>s3:PutEncryptionConfiguration</code>	Erforderlich, um die Standardverschlüsselung für einen S3-Bucket zu konfigurieren.
PutBucketIntelligentTieringConfiguration	(Erforderlich) <code>s3:PutIntelligentTieringConfiguration</code>	Erforderlich, um die S3 Intelligent-Tiering-Konfiguration in einen S3-Bucket zu übernehmen.
PutBucketInventoryConfiguration	(Erforderlich) <code>s3:PutInventoryConfiguration</code>	Erforderlich, um einem S3-Bucket eine Inventarkonfiguration hinzuzufügen.
PutBucketLifecycle	(Erforderlich) <code>s3:PutLifecycleConfiguration</code>	Erforderlich, um eine neue S3-Lifecycle-Konfiguration zu erstellen oder eine bestehende Lifecycle-Konfiguration für einen S3-Bucket zu ersetzen.
PutBucketLogging	(Erforderlich) <code>s3:PutBucketLogging</code>	Erforderlich, um die Protokollierungsparameter für einen S3-Bucket festzulegen und Berechtigungen dafür anzugeben, wer die Protokollierungsparameter anzeigen und ändern kann.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutBucketMetricsConfiguration	(Erforderlich) <code>s3:PutMetricsConfiguration</code>	Erforderlich, um eine Metrikkonfiguration für die CloudWatch Amazon-Anforderungsmetriken eines S3-Buckets festzulegen oder zu aktualisieren.
PutBucketNotificationConfiguration	(Erforderlich) <code>s3:PutBucketNotification</code>	Erforderlich, um Benachrichtigungen über bestimmte Ereignisse für einen S3-Bucket zu aktivieren.
PutBucketOwnershipControls	(Erforderlich) <code>s3:PutBucketOwnershipControls</code>	Erforderlich, um die Einstellung „Objektbesitz“ für einen S3-Bucket zu erstellen oder zu ändern.
PutBucketPolicy	(Erforderlich) <code>s3:PutBucketPolicy</code>	Erforderlich, um eine S3-Bucket-Richtlinie auf einen Bucket anzuwenden.
PutBucketReplication	(Erforderlich) <code>s3:PutReplicationConfiguration</code>	Erforderlich, um eine neue Replikationskonfiguration zu erstellen oder eine bestehende für einen S3-Bucket zu ersetzen.
PutBucketRequestPayment	(Erforderlich) <code>s3:PutBucketRequestPayment</code>	Erforderlich, um die Konfiguration der Zahlungsanfrage für einen Bucket festzulegen.
PutBucketTagging	(Erforderlich) <code>s3:PutBucketTagging</code>	Erforderlich, um einem S3-Bucket eine Reihe von Tags hinzuzufügen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutBucketVersioning	(Erforderlich) <code>s3:PutBucketVersioning</code>	Erforderlich, um den Versionsstatus eines S3-Buckets festzulegen.
PutBucketWebsite	(Erforderlich) <code>s3:PutBucketWebsite</code>	Erforderlich, um einen Bucket als Website zu konfigurieren und die Konfiguration der Website festzulegen.
PutObjectLockConfiguration	(Erforderlich) <code>s3:PutObjectLockConfiguration</code>	Erforderlich, um die Object Lock-Konfiguration in einem S3-Bucket zu platzieren.
PutPublicAccessBlock (Bucket-Ebene)	(Erforderlich) <code>s3:PutPublicAccessBlock</code>	Erforderlich, um die Konfiguration für den öffentlichen Blockzugriff für einen S3-Bucket zu erstellen oder zu ändern.

Objektoperationen und Berechtigungen

Objektoperationen sind API S3-Operationen, die auf dem Objektressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Objektoperationen in ressourcenbasierten Richtlinien (wie Bucket-Richtlinien, Zugriffspunktrichtlinien, Zugriffspunktrichtlinien für mehrere Regionen, VPC Endpunktrichtlinien) oder IAM identitätsbasierten Richtlinien angeben.

In den Richtlinien muss das Resource Element das Objekt sein. ARN Weitere Informationen zum Resource Elementformat und zu Beispielrichtlinien finden Sie unter [Objektoperationen](#).

Note

- AWS KMS Richtlinienaktionen (`kms:GenerateDataKey` und `kms:Decrypt`) gelten nur für AWS KMS Ressourcentyp und muss in IAM identitätsbasierten Richtlinien angegeben werden und AWS KMS ressourcenbasierte Richtlinien (AWS KMS wichtige Politiken).

Sie können nicht angeben AWS KMS Richtlinienaktionen in ressourcenbasierten S3-Richtlinien, wie z. B. S3-Bucket-Richtlinien.

- Wenn Sie Access Points verwenden, um den Zugriff auf Objektoperationen zu steuern, können Sie Zugriffspunktrichtlinien verwenden. Beachten Sie Folgendes, um Berechtigungen für Objektoperationen in Zugriffspunktrichtlinien zu erteilen:
 - In Zugriffspunktrichtlinien, die Berechtigungen für Objektoperationen gewähren, muss das Resource Element das Element ARNs für Objekte sein, auf die über einen Zugriffspunkt zugegriffen wird. Weitere Informationen zum Resource Elementformat und zu Beispielrichtlinien finden Sie unter [Objektoperationen in Zugriffspunktrichtlinien](#).
 - Nicht alle Objektoperationen werden von Access Points unterstützt. Weitere Informationen finden Sie unter [Zugriffspunkt-Kompatibilität mit S3-Vorgänge](#).
 - Nicht alle Objektoperationen werden von Access Points mit mehreren Regionen unterstützt. Weitere Informationen finden Sie unter [Kompatibilität von Multi-Region Access Points mit S3-Operationen](#).

Im Folgenden werden Objektoperationen und erforderliche Richtlinienaktionen zugeordnet.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
AbortMultipartUpload	(Erforderlich) s3:AbortMultipartUpload	Erforderlich, um einen mehrteiligen Upload abzuberechnen.
CompleteMultipartUpload	(Erforderlich) s3:PutObject	Erforderlich, um einen mehrteiligen Upload abzuschließen.
	(Bedingt erforderlich) kms:Decrypt	Erforderlich, wenn Sie einen mehrteiligen Upload für einen abschließen möchten AWS KMS Verschlüsseltes Objekt mit vom Kunden verwaltetem Schlüssel.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CopyObject	Für das Quellobjekt:	Für das Quellobjekt:
	(Erforderlich) Entweder <code>s3:GetObject</code> oder <code>s3:GetObjectVersion</code>	<ul style="list-style-type: none"> • <code>s3:GetObject</code> — Erforderlich, wenn Sie ein Objekt aus dem Quell-Bucket kopieren möchten, ohne es <code>versionId</code> in der Anfrage anzugeben. • <code>s3:GetObjectVersion</code> — Erforderlich, wenn Sie eine bestimmte Version eines Objekts aus dem Quell-Bucket kopieren möchten, indem Sie dies <code>versionId</code> in der Anfrage angeben.
	(Bedingt erforderlich) <code>kms:Decrypt</code>	Erforderlich, wenn Sie eine kopieren möchten AWS KMS Vom Kunden verwaltetes verschlüsseltes Schlüssel objekt aus dem Quell-Bucket.
	Für das Zielobjekt:	Für das Zielobjekt:
	(Erforderlich) <code>s3:PutObject</code>	Erforderlich, um das kopierte Objekt in den Ziel-Bucket zu legen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>s3:PutObjectAcl</code>	Erforderlich, wenn Sie das kopierte Objekt mit der Objektzugriffskontrollliste (ACL) in den Ziel-Bucket aufnehmen möchten, wenn Sie eine <code>CopyObject</code> Anfrage stellen.
	(Bedingt erforderlich) <code>s3:PutObjectTagging</code>	Erforderlich, wenn Sie das kopierte Objekt mit Objekt-Tagging im Ziel-Bucket platzieren möchten, wenn Sie eine <code>CopyObject</code> Anfrage stellen.
	(Bedingt erforderlich) <code>kms:GenerateDataKey</code>	Erforderlich, wenn Sie das kopierte Objekt mit einem verschlüsseln möchten AWS KMS Der vom Kunden verwaltete Schlüssel wird im Ziel-Bucket abgelegt.
	(Bedingt erforderlich) <code>s3:PutObjectRetention</code>	Erforderlich, wenn Sie eine Aufbewahrungskonfiguration für Object Lock für das neue Objekt festlegen möchten.
	(Bedingt erforderlich) <code>s3:PutObjectLegalHold</code>	Erforderlich, wenn Sie dem neuen Objekt eine gesetzliche Sperre zuweisen möchten.
CreateMultipartUpload	(Erforderlich) <code>s3:PutObject</code>	Erforderlich, um einen mehrteiligen Upload zu erstellen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>s3:PutObjectAcl</code>	Erforderlich, wenn Sie die Berechtigungen der Objektzugriffskontrollliste (ACL) für das hochgeladene Objekt festlegen möchten.
	(Bedingt erforderlich) <code>s3:PutObjectTagging</code>	Erforderlich, wenn Sie dem hochgeladenen Objekt Objekt-Tagging (s) hinzufügen möchten.
	(Bedingt erforderlich) <code>kms:GenerateDataKey</code>	Erforderlich, wenn Sie eine verwenden möchten AWS KMS Vom Kunden verwalteter Schlüssel zum Verschlüsseln eines Objekts, wenn Sie einen mehrteiligen Upload initiieren.
	(Bedingt erforderlich) <code>s3:PutObjectRetention</code>	Erforderlich, wenn Sie eine Object Lock-Aufbewahrungskonfiguration für das hochgeladene Objekt festlegen möchten.
	(Bedingt erforderlich) <code>s3:PutObjectLegalHold</code>	Erforderlich, wenn Sie dem hochgeladenen Objekt eine gesetzliche Sperre zuweisen möchten.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteObject	(Erforderlich) Entweder <code>s3:DeleteObject</code> oder <code>s3:DeleteObjectVersion</code>	<ul style="list-style-type: none"> • <code>s3:DeleteObject</code> — Erforderlich, wenn Sie ein Objekt entfernen möchten, ohne es <code>versionId</code> in der Anfrage anzugeben. • <code>s3:DeleteObjectVersion</code> — Erforderlich, wenn Sie eine bestimmte Version eines Objekts entfernen möchten, indem Sie dies <code>versionId</code> in der Anforderung angeben.
	(Bedingt erforderlich) <code>s3:BypassGovernanceRetention</code>	Erforderlich, wenn Sie ein Objekt löschen möchten, das durch den Governance-Modus für die Aufbewahrung von Object Lock geschützt ist.
DeleteObjects	(Erforderlich) Entweder <code>s3:DeleteObject</code> oder <code>s3:DeleteObjectVersion</code>	<ul style="list-style-type: none"> • <code>s3:DeleteObject</code> — Erforderlich, wenn Sie ein Objekt entfernen möchten, ohne es <code>versionId</code> in der Anfrage anzugeben. • <code>s3:DeleteObjectVersion</code> — Erforderlich, wenn Sie eine bestimmte Version eines Objekts entfernen möchten, indem Sie dies <code>versionId</code> in der Anforderung angeben.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>s3:BypassGovernanceRetention</code>	Erforderlich, wenn Sie Objekte löschen möchten, die durch den Governance-Modus für die Aufbewahrung von Object Lock geschützt sind.
DeleteObjectTagging	(Erforderlich) Entweder <code>s3:DeleteObjectTagging</code> oder <code>s3:DeleteObjectVersionTagging</code>	<ul style="list-style-type: none"> • <code>s3:DeleteObjectTagging</code> — Erforderlich, wenn Sie den gesamten Tagsatz eines Objekts entfernen möchten, ohne dies <code>versionId</code> in der Anforderung anzugeben. • <code>s3:DeleteObjectVersionTagging</code> — Erforderlich, wenn Sie Tags einer bestimmten Objektversion löschen möchten, indem Sie sie <code>versionId</code> in der Anfrage angeben.
GetObject	(Erforderlich) Entweder <code>s3:GetObject</code> oder <code>s3:GetObjectVersion</code>	<ul style="list-style-type: none"> • <code>s3:GetObject</code> — Erforderlich, wenn Sie ein Objekt abrufen möchten, ohne es <code>versionId</code> in der Anfrage anzugeben. • <code>s3:GetObjectVersion</code> — Erforderlich, wenn Sie eine bestimmte Version eines Objekts abrufen möchten, indem Sie es <code>versionId</code> in der Anfrage angeben.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>kms:Decrypt</code>	Erforderlich, wenn Sie eine abrufen und entschlüsseln möchten AWS KMS Vom Kunden verwaltetes verschlüsseltes Schlüsselobjekt.
	(Bedingt erforderlich) <code>s3:GetObjectTagging</code>	Erforderlich, wenn Sie das Tag-Set eines Objekts abrufen möchten, wenn Sie eine Anfrage stellen. <code>GetObject</code>
	(Bedingt erforderlich) <code>s3:GetObjectLegalHold</code>	Erforderlich, wenn Sie den aktuellen gesetzlichen Aufbewahrungstatus eines Objekts für die Objektsperre abrufen möchten.
	(Bedingt erforderlich) <code>s3:GetObjectRetention</code>	Erforderlich, wenn Sie die Aufbewahrungseinstellungen von Object Lock für ein Objekt abrufen möchten.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetObjectAcl	(Erforderlich) Entweder <code>s3:GetObjectAcl</code> oder <code>s3:GetObjectVersionAcl</code>	<ul style="list-style-type: none"> • <code>s3:GetObjectAcl</code> — Erforderlich, wenn Sie die Zugriffskontrollliste (ACL) eines Objekts abrufen möchten, ohne dies <code>versionId</code> in der Anforderung anzugeben. • <code>s3:GetObjectVersionAcl</code> — Erforderlich, wenn Sie die Zugriffskontrollliste (ACL) eines Objekts abrufen möchten, indem Sie es <code>versionId</code> in der Anforderung angeben.
GetObjectAttributes	(Erforderlich) Entweder <code>s3:GetObject</code> oder <code>s3:GetObjectVersion</code>	<ul style="list-style-type: none"> • <code>s3:GetObject</code> — Erforderlich, wenn Sie Attribute abrufen möchten, die sich auf ein Objekt beziehen, ohne dies <code>versionId</code> in der Anforderung anzugeben. • <code>s3:GetObjectVersion</code> — Erforderlich, wenn Sie Attribute abrufen möchten, die sich auf eine bestimmte Objektversion beziehen, indem Sie sie <code>versionId</code> in der Anforderung angeben.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>kms:Decrypt</code>	Erforderlich, wenn Sie Attribute abrufen möchten, die sich auf ein beziehen AWS KMS Verschlüsseltes Objekt mit vom Kunden verwaltetem Schlüssel.
GetObjectLegalHold	(Erforderlich) <code>s3:GetObjectLegalHold</code>	Erforderlich, um den aktuellen gesetzlichen Aufbewahrungstatus eines Objekts für Object Lock abzurufen.
GetObjectRetention	(Erforderlich) <code>s3:GetObjectRetention</code>	Erforderlich, um die Object Lock-Aufbewahrungseinstellungen für ein Objekt abzurufen.
GetObjectTagging	(Erforderlich) Entweder <code>s3:GetObjectTagging</code> oder <code>s3:GetObjectVersionTagging</code>	<ul style="list-style-type: none"> • <code>s3:GetObjectTagging</code> — Erforderlich, wenn Sie den Tagsatz eines Objekts ohne Angabe <code>versionId</code> in der Anfrage abrufen möchten. • <code>s3:GetObjectVersionTagging</code> — Erforderlich, wenn Sie die Tags einer bestimmten Objektversion durch Angabe <code>versionId</code> in der Anfrage abrufen möchten.
GetObjectTorrent	(Erforderlich) <code>s3:GetObject</code>	Erforderlich, um Torrent-Dateien eines Objekts zurückzugeben.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
HeadObject	(Erforderlich) <code>s3:GetObject</code>	Erforderlich, um Metadaten von einem Objekt abzurufen, ohne das Objekt selbst zurückzugeben.
	(Bedingt erforderlich) <code>s3:GetObjectLegalHold</code>	Erforderlich, wenn Sie den aktuellen gesetzlichen Aufbewahrungstatus eines Objekts für die Objektsperre abrufen möchten.
	(Bedingt erforderlich) <code>s3:GetObjectRetention</code>	Erforderlich, wenn Sie die Aufbewahrungseinstellungen von Object Lock für ein Objekt abrufen möchten.
ListMultipartUploads	(Erforderlich) <code>s3:ListBucketMultipartUploads</code>	Erforderlich, um laufende mehrteilige Uploads in einem Bucket aufzulisten.
ListParts	(Erforderlich) <code>s3:ListMultipartUploadParts</code>	Erforderlich, um die Teile aufzulisten, die für einen bestimmten mehrteiligen Upload hochgeladen wurden.
	(Bedingt erforderlich) <code>kms:Decrypt</code>	Erforderlich, wenn Sie Teile eines auflisten möchten AWS KMS Vom Kunden verwalteter, verschlüsselter, mehrteiliger Upload.
PutObject	(Erforderlich) <code>s3:PutObject</code>	Erforderlich, um ein Objekt zu platzieren.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>s3:PutObjectAcl</code>	Erforderlich, wenn Sie die Objektzugriffskontrollliste (ACL) bei einer <code>PutObject</code> Anfrage angeben möchten.
	(Bedingt erforderlich) <code>s3:PutObjectTagging</code>	Erforderlich, wenn Sie bei einer Anfrage Objekt-Tagging angeben möchten. <code>PutObject</code>
	(Bedingt erforderlich) <code>kms:GenerateDataKey</code>	Erforderlich, wenn Sie ein Objekt mit einem verschlüsseln möchten AWS KMS vom Kunden verwalteter Schlüssel.
	(Bedingt erforderlich) <code>s3:PutObjectRetention</code>	Erforderlich, wenn Sie eine Aufbewahrungskonfiguration für Object Lock für ein Objekt festlegen möchten.
	(Bedingt erforderlich) <code>s3:PutObjectLegalHold</code>	Erforderlich, wenn Sie eine Object Lock-Konfiguration mit legaler Aufbewahrungsfrist auf ein bestimmtes Objekt anwenden möchten.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutObjectAcl	(Erforderlich) Entweder <code>s3:PutObjectAcl</code> oder <code>s3:PutObjectVersionAcl</code>	<ul style="list-style-type: none"> • <code>s3:PutObjectAcl</code> — Erforderlich, wenn Sie die Zugriffssteuerungsliste (ACL) -Berechtigungen für ein neues oder vorhandenes Objekt festlegen möchten, ohne dies <code>versionId</code> in der Anforderung anzugeben. • <code>s3:PutObjectVersionAcl</code> — Erforderlich, wenn Sie die Zugriffssteuerungsliste (ACL) -Berechtigungen für ein neues oder vorhandenes Objekt durch Angabe <code>versionId</code> in der Anforderung festlegen möchten.
PutObjectLegalHold	(Erforderlich) <code>s3:PutObjectLegalHold</code>	Erforderlich, um eine Object Lock Legal Hold-Konfiguration auf ein Objekt anzuwenden.
PutObjectRetention	(Erforderlich) <code>s3:PutObjectRetention</code>	Erforderlich, um eine Object Lock-Aufbewahrungskonfiguration auf ein Objekt anzuwenden.
	(Bedingt erforderlich) <code>s3:BypassGovernanceRetention</code>	Erforderlich, wenn Sie den Verwaltungsmodus einer Object Lock-Aufbewahrungskonfiguration umgehen möchten.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutObjectTagging	(Erforderlich) Entweder <code>s3:PutObjectTagging</code> oder <code>s3:PutObjectVersionTagging</code>	<ul style="list-style-type: none"> • <code>s3:PutObjectTagging</code> — Erforderlich, wenn Sie den angegebenen Tagsatz auf ein Objekt setzen möchten, das bereits in einem Bucket vorhanden ist, ohne dass dies <code>versionId</code> in der Anforderung angegeben wurde. • <code>s3:PutObjectVersionTagging</code> — Erforderlich, wenn Sie den angegebenen Tagsatz auf ein Objekt setzen möchten, das bereits in einem Bucket vorhanden ist, indem Sie dies <code>versionId</code> in der Anfrage angeben.
RestoreObject	(Erforderlich) <code>s3:RestoreObject</code>	Erforderlich, um eine Kopie eines archivierten Objekts wiederherzustellen.
SelectObjectContent	(Erforderlich) <code>s3:GetObject</code>	Erforderlich, um den Inhalt eines S3-Objekts auf der Grundlage einer einfachen strukturierten Query Language (SQL) -Anweisung zu filtern.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>kms:Decrypt</code>	Erforderlich, wenn Sie den Inhalt eines S3-Objekts filtern möchten, das mit einem verschlüsselt ist AWS KMS vom Kunden verwalteter Schlüssel.
UploadPart	(Erforderlich) <code>s3:PutObject</code>	Erforderlich, um ein Teil in einem mehrteiligen Upload hochzuladen.
	(Bedingt erforderlich) <code>kms:GenerateDataKey</code>	Erforderlich, wenn Sie einen Upload-Teil hochladen und mit einem verschlüsseln möchten AWS KMS vom Kunden verwalteter Schlüssel.
UploadPartCopy	Für das Quellobjekt:	Für das Quellobjekt:
	(Erforderlich) Entweder <code>s3:GetObject</code> oder <code>s3:GetObjectVersion</code>	<ul style="list-style-type: none"> • <code>s3:GetObject</code> — Erforderlich, wenn Sie ein Objekt aus dem Quell-Bucket kopieren möchten, ohne es <code>versionId</code> in der Anfrage anzugeben. • <code>s3:GetObjectVersion</code> — Erforderlich, wenn Sie eine bestimmte Version eines Objekts aus dem Quell-Bucket kopieren möchten, indem Sie dies <code>versionId</code> in der Anfrage angeben.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>kms:Decrypt</code>	Erforderlich, wenn Sie eine kopieren möchten AWS KMS Vom Kunden verwaltetes verschlüsseltes Schlüssel objekt aus dem Quell-Bucket.
	Für den Zielteil:	Für den Zielteil:
	(Erforderlich) <code>s3:PutObject</code>	Erforderlich, um einen mehrteiligen Upload-Teil in den Ziel-Bucket hochzuladen.
	(Bedingt erforderlich) <code>kms:GenerateDataKey</code>	Erforderlich, wenn Sie einen Teil mit einem verschlüsseln möchten AWS KMS vom Kunden verwalteter Schlüssel , wenn Sie das Teil in den Ziel-Bucket hochladen.

Operationen und Berechtigungen des Access Points

Access Point-Operationen sind API S3-Operationen, die auf dem `accesspoint` Ressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Access Point-Operationen in IAM identitätsbasierten Richtlinien angeben, nicht in Bucket-Richtlinien oder Zugriffspunktrichtlinien.

In den Richtlinien muss das `Resource` Element das sein. `accesspoint` ARN Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unter [Operationen an Zugriffspunkten](#).

Note

Wenn Sie Access Points verwenden möchten, um den Zugriff auf Bucket- oder Objektoperationen zu steuern, beachten Sie Folgendes:

- Informationen zur Verwendung von Access Points zur Steuerung des Zugriffs auf Bucket-Operationen finden Sie unter [Bucket-Operationen in Access Point-Richtlinien](#).
- Informationen zur Verwendung von Access Points zur Steuerung des Zugriffs auf Objektoperationen finden Sie unter [Objektoperationen in Zugriffspunktrichtlinien](#).
- Weitere Informationen zur Konfiguration von Zugriffspunktrichtlinien finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#).

Im Folgenden werden die Zugriffspunktvorgänge und die erforderlichen Richtlinienaktionen zugeordnet.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateAccessPoint	(Erforderlich) s3:CreateAccessPoint	Erforderlich, um einen Access Point zu erstellen, der einem S3-Bucket zugeordnet ist.
DeleteAccessPoint	(Erforderlich) s3:DeleteAccessPoint	Erforderlich, um einen Access Point zu löschen.
DeleteAccessPointPolicy	(Erforderlich) s3:DeleteAccessPointPolicy	Erforderlich, um eine Zugriffspunktrichtlinie zu löschen.
GetAccessPointPolicy	(Erforderlich) s3:GetAccessPointPolicy	Erforderlich, um eine Zugriffspunktrichtlinie abzurufen.
GetAccessPointPolicyStatus	(Erforderlich) s3:GetAccessPointPolicyStatus	Erforderlich, um die Informationen darüber abzurufen, ob für den angegebenen Zugriffspunkt derzeit eine Richtlinie gilt, die den öffentlichen Zugriff ermöglicht.
PutAccessPointPolicy	(Erforderlich) s3:PutAccessPointPolicy	Erforderlich, um eine Zugriffspunktrichtlinie festzulegen.

Operationen und Berechtigungen für Object Lambda Access Point

Object Lambda Access Point-Operationen sind API S3-Operationen, die auf dem `objectlambdaaccesspoint` Ressourcentyp ausgeführt werden. Weitere Informationen zur Konfiguration von Richtlinien für Object Lambda Access Point-Operationen finden Sie unter [Konfiguration von IAM Richtlinien für Object Lambda Access Points](#).

Im Folgenden finden Sie die Zuordnung der Object Lambda Access Point-Operationen und der erforderlichen Richtlinienaktionen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateAccessPointForObjectLambda	(Erforderlich) <code>s3:CreateAccessPointForObjectLambda</code>	Erforderlich, um einen Object Lambda Access Point zu erstellen.
DeleteAccessPointForObjectLambda	(Erforderlich) <code>s3:DeleteAccessPointForObjectLambda</code>	Erforderlich, um einen angegebenen Object Lambda Access Point zu löschen.
DeleteAccessPointPolicyForObjectLambda	(Erforderlich) <code>s3:DeleteAccessPointPolicyForObjectLambda</code>	Erforderlich, um die Richtlinie auf einem angegebenen Object Lambda Access Point zu löschen.
GetAccessPointConfigurationForObjectLambda	(Erforderlich) <code>s3:GetAccessPointConfigurationForObjectLambda</code>	Erforderlich, um die Konfiguration des Object Lambda Access Points abzurufen.
GetAccessPointForObjectLambda	(Erforderlich) <code>s3:GetAccessPointForObjectLambda</code>	Erforderlich, um Informationen über den Object Lambda Access Point abzurufen.
GetAccessPointPolicyForObjectLambda	(Erforderlich) <code>s3:GetAccessPointPolicyForObjectLambda</code>	Erforderlich, um die Zugriffspunktrichtlinie zurückzugeben, die dem angegebenen Object Lambda Access Point zugeordnet ist.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetAccessPointPolicyStatusForObjectLambda	(Erforderlich) <code>s3:GetAccessPointPolicyStatusForObjectLambda</code>	Erforderlich, um den Richtlinienstatus für eine bestimmte Object Lambda Access Point-Richtlinie zurückzugeben.
PutAccessPointConfigurationForObjectLambda	(Erforderlich) <code>s3:PutAccessPointConfigurationForObjectLambda</code>	Erforderlich, um die Konfiguration des Object Lambda Access Points festzulegen.
PutAccessPointPolicyForObjectLambda	(Erforderlich) <code>s3:PutAccessPointPolicyForObjectLambda</code>	Erforderlich, um einem angegebenen Object Lambda Access Point eine Zugriffsrichtlinie zuzuordnen.

Operationen und Berechtigungen für Access Points in mehreren Regionen

Access Point-Operationen mit mehreren Regionen sind API S3-Operationen, die auf dem `multiregionaccesspoint` Ressourcentyp ausgeführt werden. Weitere Informationen zur Konfiguration von Richtlinien für Access Point-Operationen in mehreren Regionen finden Sie unter [Beispielrichtlinien für Multi-Region Access Points](#)

Im Folgenden finden Sie die Zuordnung von Access Point-Vorgängen in mehreren Regionen und den erforderlichen Richtlinienaktionen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateMultiRegionAccessPoint	(Erforderlich) <code>s3:CreateMultiRegionAccessPoint</code>	Erforderlich, um einen Access Point mit mehreren Regionen zu erstellen und ihn mit S3-Buckets zu verknüpfen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteMultiRegionAccessPoint	(Erforderlich) <code>s3:DeleteMultiRegionAccessPoint</code>	Erforderlich, um einen Access Point mit mehreren Regionen zu löschen.
DescribeMultiRegionAccessPointOperation	(Erforderlich) <code>s3:DescribeMultiRegionAccessPointOperation</code>	Erforderlich, um den Status einer asynchronen Anfrage zur Verwaltung eines Access Points mit mehreren Regionen abzurufen.
GetMultiRegionAccessPoint	(Erforderlich) <code>s3:GetMultiRegionAccessPoint</code>	Erforderlich, um Konfigurationsinformationen über den angegebenen Multi-Region-Access Point zurückzugeben.
GetMultiRegionAccessPointPolicy	(Erforderlich) <code>s3:GetMultiRegionAccessPointPolicy</code>	Erforderlich, um die Zugriffskontrollrichtlinie des angegebenen Multi-Region-Access Points zurückzugeben.
GetMultiRegionAccessPointPolicyStatus	(Erforderlich) <code>s3:GetMultiRegionAccessPointPolicyStatus</code>	Erforderlich, um den Richtlinienstatus für einen bestimmten Multi-Region-Access Point zurückzugeben, der angibt, ob der angegebene Multi-Region-Access Point über eine Zugriffskontrollrichtlinie verfügt, die öffentlichen Zugriff ermöglicht.
GetMultiRegionAccessPointRoutes	(Erforderlich) <code>s3:GetMultiRegionAccessPointRoutes</code>	Erforderlich, um die Routing-Konfiguration für einen Access Point mit mehreren Regionen zurückzugeben.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutMultiRegionAccessPointPolicy	(Erforderlich) <code>s3:PutMultiRegionAccessPointPolicy</code>	Erforderlich, um die Zugriffskontrollrichtlinie des angegebenen Multi-Region-Access Points zu aktualisieren.
SubmitMultiRegionAccessPointRoutes	(Erforderlich) <code>s3:SubmitMultiRegionAccessPointRoutes</code>	Erforderlich, um eine aktualisierte Routenkonfiguration für einen Access Point mit mehreren Regionen einzureichen.

Batch-Job-Operationen und Berechtigungen

Auftragsoperationen (Batch Operations) sind API S3-Operationen, die auf dem `job` Ressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Auftragsoperationen in IAM identitätsbasierten Richtlinien angeben, nicht in Bucket-Richtlinien.

In den Richtlinien muss das `Resource` Element das sein. `job` ARN Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unter [Batch-Job-Operationen](#).

Im Folgenden werden Batch-Job-Operationen und erforderliche Richtlinienaktionen zugeordnet.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteJobTagging	(Erforderlich) <code>s3:DeleteJobTagging</code>	Erforderlich, um Tags aus einem vorhandenen S3 Batch Operations-Job zu entfernen.
DescribeJob	(Erforderlich) <code>s3:DescribeJob</code>	Erforderlich, um die Konfigurationsparameter und den Status für einen Batch Operations-Job abzurufen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetJobTagging	(Erforderlich) s3:GetJobTagging	Erforderlich, um den Tagsatz eines vorhandenen S3 Batch Operations-Jobs zurückzugeben.
PutJobTagging	(Erforderlich) s3:PutJobTagging	Erforderlich, um einem vorhandenen S3 Batch Operations-Job Tags hinzuzufügen oder zu ersetzen.
UpdateJobPriority	(Erforderlich) s3:UpdateJobPriority	Erforderlich, um die Priorität eines vorhandenen Jobs zu aktualisieren.
UpdateJobStatus	(Erforderlich) s3:UpdateJobStatus	Erforderlich, um den Status für den angegebenen Job zu aktualisieren.

Konfiguration, Vorgänge und Berechtigungen von S3 Storage Lens

S3 Storage Lens-Konfigurationsvorgänge sind API S3-Operationen, die auf dem `storage-lens-configuration` Ressourcentyp ausgeführt werden. Weitere Informationen zur Konfiguration von S3 Storage Lens-Konfigurationsvorgängen finden Sie unter [Berechtigungen für Amazon S3 Storage Lens](#).

Im Folgenden werden die Konfigurationsvorgänge von S3 Storage Lens und die erforderlichen Richtlinienaktionen zugeordnet.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteStorageLensConfiguration	(Erforderlich) s3:DeleteStorageLensConfiguration	Erforderlich, um die S3 Storage Lens-Konfiguration zu löschen.
DeleteStorageLensConfigurationTagging	(Erforderlich) s3:DeleteStorageLensConfigurationTagging	Erforderlich, um die S3 Storage Lens-Konfigurations-Tags zu löschen.
GetStorageLensConfiguration	(Erforderlich) s3:GetStorageLensConfiguration	Erforderlich, um die S3 Storage Lens-Konfiguration abzurufen.
GetStorageLensConfigurationTagging	(Erforderlich) s3:GetStorageLensConfigurationTagging	Erforderlich, um die Tags der S3 Storage Lens-Konfiguration abzurufen.
PutStorageLensConfigurationTagging	(Erforderlich) s3:PutStorageLensConfigurationTagging	Erforderlich, um Tags in eine bestehende S3-Speicherlinsenkonfiguration einzufügen oder zu ersetzen.

S3 Storage Lens gruppiert Operationen und Berechtigungen

Bei Operationen mit S3 Storage Lens-Gruppen handelt es sich um API S3-Operationen, die auf dem `storagelensgroup` Ressourcentyp ausgeführt werden. Weitere Informationen zur Konfiguration von Gruppenberechtigungen für S3 Storage Lens finden Sie unter [Berechtigungen für Storage-Lens-Gruppen](#).

Im Folgenden finden Sie die Zuordnung von S3 Storage Lens-Gruppenoperationen und erforderlichen Richtlinienaktionen.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteStorageLensGroup	(Erforderlich) <code>s3:DeleteStorageLensGroup</code>	Erforderlich, um eine bestehende S3 Storage Lens-Gruppe zu löschen.
GetStorageLensGroup	(Erforderlich) <code>s3:GetStorageLensGroup</code>	Erforderlich, um die Konfigurationsdetails der S3 Storage Lens-Gruppe abzurufen.
UpdateStorageLensGroup	(Erforderlich) <code>s3:UpdateStorageLensGroup</code>	Erforderlich, um die bestehende S3 Storage Lens-Gruppe zu aktualisieren.

Kontooperationen und Berechtigungen

Kontovorgänge sind API S3-Operationen, die auf Kontoebene ausgeführt werden. Konto ist kein von Amazon S3 definierter Ressourcentyp. Sie müssen S3-Richtlinienaktionen für Kontooperationen in IAM identitätsbasierten Richtlinien angeben, nicht in Bucket-Richtlinien.

In den Richtlinien muss das Resource Element sein. "*" Weitere Informationen zu Beispielen Richtlinien finden Sie unter [Kontovorgänge](#).

Im Folgenden werden Kontovorgänge und erforderliche Richtlinienaktionen zugeordnet.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateJob	(Erforderlich) <code>s3:CreateJob</code>	Erforderlich, um einen neuen S3 Batch Operations-Job zu erstellen.
CreateStorageLensGroup	(Erforderlich) <code>s3:CreateStorageLensGroup</code>	Erforderlich, um eine neue S3 Storage Lens-Gruppe zu erstellen und sie der angegebenen zuzuordnen AWS-Konto ID.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) <code>s3:TagResource</code>	Erforderlich, wenn Sie eine S3 Storage Lens-Gruppe mit erstellen möchten AWS Ressourcen-Tags.
<u>DeletePublicAccessBlock</u> (Auf Kontoebene)	(Erforderlich) <code>s3:PutAccountPublicAccessBlock</code>	Erforderlich, um die Konfiguration zum Blockieren des öffentlichen Zugriffs aus einem zu entfernen AWS-Konto.
<u>GetAccessPoint</u>	(Erforderlich) <code>s3:GetAccessPoint</code>	Erforderlich, um Konfigurationsinformationen über den angegebenen Zugriffspunkt abzurufen.
<u>GetAccessPointPolicy</u> (Auf Kontoebene)	(Erforderlich) <code>s3:GetAccountPublicAccessBlock</code>	Erforderlich, um die Konfiguration für den öffentlichen Blockzugriff für einen abzurufen AWS-Konto.
<u>ListAccessPoints</u>	(Erforderlich) <code>s3:ListAccessPoints</code>	Erforderlich, um Access Points eines S3-Buckets aufzulisten, die einem gehören AWS-Konto.
<u>ListAccessPointsForObjectLambda</u>	(Erforderlich) <code>s3:ListAccessPointsForObjectLambda</code>	Erforderlich, um die Object Lambda Access Points aufzulisten.
<u>ListBuckets</u>	(Erforderlich) <code>s3:ListAllMyBuckets</code>	Erforderlich, um eine Liste aller Buckets zurückzugeben, die dem authentifizierten Absender der Anfrage gehören.

API Operationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
ListJobs	(Erforderlich) <code>s3:ListJobs</code>	Erforderlich, um aktuelle Jobs und Jobs aufzulisten, die kürzlich beendet wurden.
ListMultiRegionAccessPoints	(Erforderlich) <code>s3:ListMultiRegionAccessPoints</code>	Erforderlich, um eine Liste der Multi-Region-Access Points zurückzugeben, die derzeit mit den angegebenen Access Points verknüpft sind AWS-Konto.
ListStorageLensConfigurations	(Erforderlich) <code>s3:ListStorageLensConfigurations</code>	Erforderlich, um eine Liste der S3 Storage Lens-Konfigurationen für ein abzurufen AWS-Konto.
ListStorageLensGroups	(Erforderlich) <code>s3:ListStorageLensGroups</code>	Erforderlich, um alle S3 Storage Lens-Gruppen im angegebenen Zuhause aufzulisten AWS-Region.
PutPublicAccessBlock(Kontoebene)	(Erforderlich) <code>s3:PutAccountPublicAccessBlock</code>	Erforderlich, um die Konfiguration für den öffentlichen Zugriff blockieren für einen zu erstellen oder zu ändern AWS-Konto.
PutStorageLensConfiguration	(Erforderlich) <code>s3:PutStorageLensConfiguration</code>	Erforderlich, um eine S3-Speicherlinsenkonfiguration einzurichten.

Richtlinien und Berechtigungen in Amazon S3

Diese Seite bietet einen Überblick über Bucket- und Benutzerrichtlinien in Amazon S3 und beschreibt die grundlegenden Elemente eines AWS Identity and Access Management (IAM) Richtlinie. Jedes

aufgelistete Element verweist auf weitere Details zu diesem Element und auf Beispiele für die Verwendung dieses Elements.

Eine vollständige Liste der Amazon S3-Aktionen, Ressourcen und Bedingungen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

In ihrer einfachsten Form enthält eine Richtlinie die folgenden Elemente:

- [Resource](#)— Der Amazon S3 S3-Bucket, das Objekt, der Access Point oder der Job, für den die Richtlinie gilt. Verwenden Sie den Amazon-Ressourcennamen (ARN) des Buckets, Objekts, Access Points oder Jobs, um die Ressource zu identifizieren.

Ein Beispiel für Operationen auf Bucket-Ebene:

```
"Resource": "arn:aws:s3:::bucket_name"
```

Beispiele für Operationen auf Objektebene:

- "Resource": "arn:aws:s3:::*bucket_name*/*" für alle Objekte im Bucket.
- "Resource": "arn:aws:s3:::*bucket_name/prefix*/*" für Objekte unter einem bestimmten Präfix im Bucket.

Weitere Informationen finden Sie unter [Richtlinienressourcen für Amazon S3](#).

- [Actions](#)— Für jede Ressource unterstützt Amazon S3 eine Reihe von Vorgängen. Sie identifizieren RessourcenVorgänge, die Sie zulassen (oder ablehnen) können, indem Sie Aktionsschlüsselwörter verwenden.

Die `s3:ListBucket` Berechtigung ermöglicht es dem Benutzer beispielsweise, Amazon S3 zu verwenden. [ListObjectsV2](#) Betrieb. (Bei der `s3:ListBucket` Berechtigung handelt es sich um einen Fall, in dem der Aktionsname nicht direkt dem Operationsnamen zugeordnet ist.) Weitere Informationen zur Verwendung von Amazon-S3-Aktionen finden Sie unter [Politische Maßnahmen für Amazon S3](#). Eine vollständige Liste der Amazon S3 S3-Aktionen finden Sie unter [Aktionen](#) in der Amazon Simple Storage API Service-Referenz.

- [Effect](#)— Welche Auswirkung hat, wenn der Benutzer die spezifische Aktion anfordert — dies kann entweder `Allow` sein. `Deny`

Wenn Sie den Zugriff auf eine Ressource nicht ausdrücklich gestatten („Allow“), wird er automatisch verweigert. Sie können den Zugriff auf eine Ressource auch explizit verweigern. So könnten Sie z. B. sicherstellen, dass ein Benutzer nicht auf die Ressource zugreifen kann, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird. Weitere Informationen finden Sie unter [IAMJSONPolicy Elements: Effect](#) im IAMBenutzerhandbuch.

- **Principal**— Das Konto oder der Benutzer, dem Zugriff auf die Aktionen und Ressourcen in der Erklärung gewährt wird. In einer Bucket-Richtlinie ist der Prinzipal der Benutzer, das Konto, der Service oder eine andere Entität, der/die/das der Empfänger dieser Berechtigung ist. Weitere Informationen finden Sie unter [Prinzipien für Bucket-Richtlinien](#).
- **Condition**— Bedingungen für das Inkrafttreten einer Richtlinie. Sie können Folgendes verwenden ... AWS-weite Schlüssel und Amazon S3-spezifische Schlüssel zur Festlegung von Bedingungen in einer Amazon S3 S3-Zugriffsrichtlinie. Weitere Informationen finden Sie unter [Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln](#).

Die folgende Beispiel-Bucket-Richtlinie zeigt die Elemente `Effect`, `Principal`, `Action` und `Resource`. Diese Richtlinie ermöglicht *Akua* einem Benutzer im Konto `123456789012`, `s3:GetObjects`, `s3:GetBucketLocation`, und `s3:ListBucket` Amazon S3 S3-Berechtigungen für den `amzn-s3-demo-bucket1` Bucket.

```
{
  "Version": "2012-10-17",
  "Id": "ExamplePolicy01",
  "Statement": [
    {
      "Sid": "ExampleStatement01",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Akua"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1/*",
        "arn:aws:s3:::amzn-s3-demo-bucket1"
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

Vollständige Informationen zur Sprache der Richtlinie finden Sie unter [Richtlinien und Berechtigungen IAM](#) und [Referenzinformationen zu IAM JSON Richtlinien](#) im IAMBenutzerhandbuch.

Berechtigungsdelegation

Wenn ein AWS-Konto besitzt eine Ressource, kann sie diese Berechtigungen einer anderen Person gewähren AWS-Konto. Dieses Konto kann dann diese Berechtigungen oder einen Teil davon an Benutzer im Konto delegieren. Dies wird als Delegation von Berechtigungen bezeichnet. Ein Konto, das Berechtigungen von einem anderen Konto erhält, kann jedoch keine kontoübergreifenden Berechtigungen an ein anderes delegieren AWS-Konto.

Amazon-S3-Bucket- und Objekt-Eigentümerschaft

Buckets und Objekte sind Amazon-S3-Ressourcen. Standardmäßig kann nur der jeweilige Eigentümer auf diese Ressourcen zugreifen. Der Besitzer der Ressource bezieht sich auf AWS-Konto das erstellt die Ressource. Beispielsweise:

- Das Tool AWS-Konto die Sie verwenden, um Buckets zu erstellen und Objekte hochzuladen, besitzt diese Ressourcen.
- Wenn Sie ein Objekt hochladen mit AWS Identity and Access Management (IAM) Benutzer- oder Rollenmeldedaten, die AWS-Konto dem der Benutzer oder die Rolle angehört, besitzt das Objekt.
- Ein Bucket-Besitzer kann einem anderen Benutzer kontoübergreifende Berechtigungen gewähren AWS-Konto (oder Benutzern in einem anderen Konto), um Objekte hochzuladen. In diesem Fall gibt das Feld AWS-Konto derjenige, der Objekte hochlädt, besitzt diese Objekte. Der Bucket-Besitzer hat keine Berechtigungen für die Objekte, die anderen Konten gehören, mit den folgenden Ausnahmen:
 - Der Bucket-Eigentümer zahlt die Rechnungen. Ein Bucket-Eigentümer kann explizit den Zugriff auf Objekte verweigern oder Objekte im Bucket löschen, unabhängig davon, wem sie gehören.
 - Ein Bucket-Eigentümer kann Objekte archivieren oder archivierte Objekte wiederherstellen, unabhängig davon, wem sie gehören. Die Archivierung bezieht sich auf die Speicherklasse, die beim Speichern der Objekte verwendet wurde. Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Eigentümerschaft und Anforderungsauthentifizierung

Alle Anforderungen nach einem Bucket sind authentifiziert oder nicht authentifiziert. Authentifizierte Anforderungen müssen einen Signaturwert enthalten, der den Absender der Anforderung authentifiziert, und für nicht authentifizierte Anforderungen gilt dies nicht. Weitere Informationen über Anforderungsauthentifizierung finden Sie unter [Senden von Anforderungen](#).

Der Eigentümer eines Buckets kann nicht authentifizierte Anforderungen zulassen. Zum Beispiel nicht authentifiziert [PutObject](#)Anfragen sind zulässig, wenn für einen Bucket eine öffentliche Bucket-Richtlinie gilt oder wenn ein Bucket speziell der All Users Gruppe WRITE oder dem anonymen Benutzer FULL_CONTROL Zugriff ACL gewährt. Weitere Informationen zu öffentlichen Bucket-Richtlinien und öffentlichen Zugriffskontrolllisten (ACLs) finden Sie unter [Die Bedeutung von „öffentlich“](#).

Alle nicht authentifizierten Anforderungen werden vom anonymen Benutzer erstellt. Dieser Benutzer wird ACLs durch die spezifische kanonische Benutzer-ID repräsentiert. 65a011a29cdf8ec533ec3d1c0aae921c Wenn ein Objekt mit einer nicht authentifizierten Anforderung zu einem Bucket hochgeladen wird, ist der anonyme Benutzer Eigentümer des Objekts. Das Standardobjekt ACL gewährt FULL_CONTROL dem anonymen Benutzer die Rolle des Eigentümers des Objekts. Daher erlaubt Amazon S3 nicht authentifizierte Anfragen, um das Objekt abzurufen oder zu ändern. ACL

Um zu verhindern, dass Objekte vom anonymen Benutzer geändert werden, empfehlen wir, keine Bucket-Richtlinien zu implementieren, die anonyme öffentliche Schreibvorgänge in Ihren Bucket zulassen, oder solche zu verwenden ACLs, die anonymen Benutzern Schreibzugriff auf Ihren Bucket gewähren. Sie können diese empfohlene Verhaltensweise durch die Verwendung von Amazon S3 Block Public Access erzwingen.

Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#). Weitere Informationen zu finden ACLs Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Important

Wir empfehlen, dass Sie das nicht verwenden AWS-Konto Root-Benutzeranmeldedaten, um authentifizierte Anfragen zu stellen. Erstellen Sie stattdessen eine IAM Rolle und gewähren Sie dieser Rolle vollen Zugriff. Wir bezeichnen Benutzer mit dieser Rolle als Administratorbenutzer. Sie können anstelle von Anmeldeinformationen, die der Administratorrolle zugewiesen sind, verwenden AWS-Konto Root-Benutzeranmeldedaten, mit

denen Sie interagieren AWS und Aufgaben wie das Erstellen eines Buckets, das Erstellen von Benutzern und das Erteilen von Berechtigungen ausführen. Weitere Informationen finden Sie unter [AWS Sicherheitsnachweise](#) und [bewährte Sicherheitsmethoden finden Sie IAM im IAM Benutzerhandbuch](#).

Bucket-Richtlinien für Amazon S3

Eine Bucket-Richtlinie ist eine auf Ressourcen basierende Richtlinie, die Sie verwenden können, um Zugriffsberechtigungen für Ihren Amazon-S3-Bucket und die darin enthaltenen Objekte zu erteilen. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Diese Berechtigungen gelten nicht für Objekte, die anderen gehören AWS-Konten.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen wurden, und um Zugriffskontrolllisten zu deaktivieren oder zu aktivieren (ACLs). Standardmäßig ist für Object Ownership die Einstellung Bucket Owner enforced festgelegt, und alle sind ACLs deaktiviert. Der Bucket-Eigentümer besitzt alle Objekte im Bucket und verwaltet den Datenzugriff ausschließlich mithilfe von Richtlinien.

Bucket-Richtlinien basieren auf der Verwendung JSON AWS Identity and Access Management (IAM) Sprache der Richtlinie. Sie können Bucket-Richtlinien verwenden, um Berechtigungen für die Objekte in einem Bucket hinzuzufügen oder zu verweigern. Bucket-Richtlinien können Anforderungen basierend auf den Elementen in der Richtlinie erlauben oder verweigern. Diese Elemente umfassen den Anforderer, S3-Aktionen, Ressourcen und Aspekte oder Bedingungen der Anforderung (z. B. die IP-Adresse, die für die Anforderung verwendet wird).

Sie können z. B. eine Bucket-Richtlinie erstellen, die Folgendes bewirkt:

- Erteilt anderen Konten kontoübergreifende Berechtigungen für das Hochladen von Objekten in Ihren S3-Bucket
- Stellt sicher, dass Sie als Bucket-Eigentümer die volle Kontrolle über die hochgeladenen Objekte haben

Weitere Informationen finden Sie unter [Beispiele für Amazon S3 S3-Bucket-Richtlinien](#).

⚠ Important

Sie können keine Bucket-Richtlinie verwenden, um Löschungen oder Übergänge durch eine [S3-Lebenszyklusregel](#) zu verhindern. Selbst wenn Ihre Bucket-Richtlinie beispielsweise alle Aktionen für alle Principals ablehnt, funktioniert Ihre S3-Lifecycle-Konfiguration weiterhin wie gewohnt.

Die Themen in diesem Abschnitt enthalten Beispiele und zeigen Ihnen, wie Sie in der S3-Konsole eine Bucket-Richtlinie hinzufügen können. Informationen zu identitätsbasierten Richtlinien finden Sie unter [Identitätsbasierte Richtlinien für Amazon S3](#). Weitere Informationen zur Sprache der Bucket-Richtlinie finden Sie unter [Richtlinien und Berechtigungen in Amazon S3](#).

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#)

Themen

- [Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole](#)
- [Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien](#)
- [Beispiele für Amazon S3 S3-Bucket-Richtlinien](#)
- [Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln](#)

Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole

Sie können das verwenden [AWS Policy Generator](#) und die Amazon S3 S3-Konsole, um eine neue Bucket-Richtlinie hinzuzufügen oder eine bestehende Bucket-Richtlinie zu bearbeiten. Eine Bucket-Richtlinie ist ressourcenbasiert AWS Identity and Access Management (IAM) Richtlinie. Sie fügen einem Bucket eine Bucket-Richtlinie hinzu, um andere zu gewähren AWS-Konten oder IAM Benutzer Zugriffsberechtigungen für den Bucket und die darin enthaltenen Objekte. Objektberechtigungen gelten nur für die Objekte, die der Bucket-Eigentümer erstellt. Weitere Informationen zu Bucket-Richtlinien finden Sie unter [Identity and Access Management für Amazon S3](#).

Stellen Sie sicher, dass Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge von AWS Identity and Access Management Access Analyzer bevor Sie Ihre Richtlinie speichern. IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM [Richtliniengrammatik](#) und der [bewährten Methoden](#) zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten umsetzbare Empfehlungen, die Sie beim Erstellen von Richtlinien

unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen. Weitere Informationen zur Überprüfung von Richtlinien mithilfe von IAM Access Analyzer finden Sie unter [IAMAccess Analyzer-Richtliniengültigkeit](#) im IAM-Benutzerhandbuch. Eine Liste der Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM [IAMAccess Analyzer-Referenz zur Richtlinienüberprüfung](#).

Hinweise zur Behebung von Fehlern mit einer Richtlinie finden Sie unter [Problemlösung bei Zugriff verweigert \(403 Forbidden\) Fehler in Amazon S3](#).


Eine Bucket-Richtlinie erstellen oder bearbeiten

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Bucket-Richtlinie erstellen wollen oder dessen Bucket-Richtlinie Sie bearbeiten wollen.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie unter Bucket-Richtlinie Bearbeiten aus. Die Seite Edit bucket policy (Bucket-Richtlinie bearbeiten) wird angezeigt.
6. Führen Sie auf der Seite Edit bucket policy (Bucket-Richtlinie bearbeiten) einen der folgenden Schritte aus:
 - Um Beispiele für Bucket-Richtlinien zu sehen, wählen Sie Richtlinienbeispiele. Oder schauen Sie [Beispiele für Amazon S3 S3-Bucket-Richtlinien](#) im Amazon S3 S3-Benutzerhandbuch nach.
 - Um eine Richtlinie automatisch zu generieren oder zu bearbeiten, wählen Sie JSON im Abschnitt Richtlinie die Option Richtliniengenerator.

Wenn Sie den Richtliniengenerator wählen, AWS Der Richtliniengenerator wird in einem neuen Fenster geöffnet.


- a. Auf dem AWS Wählen Sie auf der Seite Policy Generator für Select Type of Policy die Option S3 Bucket Policy aus.
- b. Fügen Sie eine Anweisung hinzu, indem Sie die Informationen in die bereitgestellten Felder eingeben, und wählen Sie dann Anweisung hinzufügen. Wiederholen Sie diesen Vorgang für so viele Anweisungen, wie Sie hinzufügen möchten. Weitere Informationen

zu diesen Feldern finden Sie in der [Referenz zu den IAM JSON Richtlinienelementen](#) im IAMBenutzerhandbuch.

 Note

Der Einfachheit halber wird auf der Seite Bucket-Richtlinie bearbeiten der Bucket ARN (Amazon-Ressourcenname) des aktuellen Buckets über dem Policy-Textfeld angezeigt. Sie können dies ARN zur Verwendung in den Kontoauszügen auf der AWS Seite „Policy Generator“.

- c. Wenn Sie mit dem Hinzufügen von Anweisungen fertig sind, wählen Sie Generieren von Richtlinien.
 - d. Kopieren Sie den generierten Richtlinien text, wählen Sie Schließen und kehren Sie zur Seite Bucket-Richtlinie bearbeiten in der Amazon-S3-Konsole zurück.
7. Bearbeiten Sie im Feld Richtlinie die bestehende Richtlinie oder fügen Sie die Bucket-Richtlinie aus dem AWS Richtlinien-Generator. Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge bevor Sie Ihre Richtlinie speichern.

 Note

Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt.

8. (Optional) Wählen Sie unten rechts Preview external access (Vorschau des externen Zugriffs) aus, um eine Vorschau anzuzeigen, wie sich Ihre neue Richtlinie auf den öffentlichen und kontoübergreifenden Zugriff auf Ihre Ressource auswirkt. Bevor Sie Ihre Richtlinie speichern, können Sie überprüfen, ob sie neue IAM Access Analyzer-Ergebnisse einführt oder bestehende Ergebnisse behebt. Wenn Sie keinen aktiven Analyzer sehen, wählen Sie Go to Access Analyzer, um einen Account Analyzer [in IAM Access Analyzer zu erstellen](#). Weitere Informationen finden Sie unter [Vorschauzugriff](#) im IAMBenutzerhandbuch.
9. Wählen Sie Save changes (Änderungen speichern) aus, wodurch Sie zur Registerkarte Permissions (Berechtigungen) zurückkehren.

Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien

Sie können Amazon S3 S3-Bucket-Richtlinien verwenden, um den Zugriff auf Buckets von bestimmten Virtual Private Cloud (VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dieser Abschnitt enthält Beispiel-Bucket-Richtlinien, mit denen Sie den Amazon S3 S3-Bucket-Zugriff von

VPC Endpunkten aus steuern können. Informationen zum Einrichten von VPC Endpunkten finden Sie unter [VPCEndgeräte](#) im VPC Benutzerhandbuch.

A VPC ermöglicht Ihnen den Start AWS Ressourcen in ein von Ihnen definiertes virtuelles Netzwerk. Ein VPC Endpunkt ermöglicht es Ihnen, eine private Verbindung zwischen Ihrem VPC und einem anderen herzustellen AWS-Service. Für diese private Verbindung ist kein Zugriff über das Internet, über eine virtuelle private Netzwerkverbindung (VPN), über eine NAT Instanz oder über AWS Direct Connect.

Ein VPC Endpunkt für Amazon S3 ist eine logische Einheit innerhalb einerVPC, die nur Konnektivität zu Amazon S3 ermöglicht. Der VPC Endpunkt leitet Anfragen an Amazon S3 weiter und leitet Antworten zurück an denVPC. VPCEndpunkte ändern nur die Art und Weise, wie Anfragen weitergeleitet werden. Öffentliche Endpunkte und DNS Namen von Amazon S3 funktionieren weiterhin mit VPC Endpunkten. Wichtige Informationen zur Verwendung von VPC Endpunkten mit Amazon S3 finden Sie unter [Gateway-Endpunkte](#) und [Gateway-Endpunkte für Amazon S3](#) im VPC Benutzerhandbuch.

VPCEndpunkte für Amazon S3 bieten zwei Möglichkeiten, den Zugriff auf Ihre Amazon S3 S3-Daten zu kontrollieren:

- Sie können die Anfragen, Benutzer oder Gruppen steuern, die über einen bestimmten VPC Endpunkt zugelassen werden. Informationen zu dieser Art der Zugriffskontrolle finden Sie im Benutzerhandbuch unter [Steuern des Zugriffs auf VPC Endgeräte mithilfe von VPC Endpunktrichtlinien](#).
- Mithilfe der Amazon S3 S3-Bucket-Richtlinien können Sie kontrollieren, welche VPCs oder welche VPC Endpunkte Zugriff auf Ihre Buckets haben. Beispiele für diese Art Zugriffssteuerung durch Bucket-Richtlinienfinden Sie in den folgenden Themen zur Zugriffsbeschränkung.

Themen

- [Beschränken des Zugriffs auf einen bestimmten Endpunkt VPC](#)
- [Beschränken Sie den Zugriff auf einen bestimmten VPC](#)

Important

Wenn Sie die in diesem Abschnitt beschriebenen Amazon S3 S3-Bucket-Richtlinien für VPC Endgeräte anwenden, können Sie Ihren Zugriff auf den Bucket unbeabsichtigt blockieren. Bucket-Berechtigungen, mit denen der Bucket-Zugriff speziell auf Verbindungen beschränkt

werden soll, die von Ihrem VPC Endpunkt ausgehen, können alle Verbindungen zum Bucket blockieren. Informationen zur Behebung dieses Problems finden Sie unter [Wie behebe ich meine Bucket-Richtlinie, wenn sie die falsche VPC oder die VPC Endpunkt-ID hat?](#) in der AWS Support Wissenszentrum.

Beschränken des Zugriffs auf einen bestimmten Endpunkt VPC

Im Folgenden finden Sie ein Beispiel für eine Amazon S3 S3-Bucket-Richtlinie, die den Zugriff auf einen bestimmten Bucket nur von dem VPC Endpunkt mit der ID `vpce-1a2b3c4d` einschränkt. `awsexamplebucket1` Wenn der angegebene Endpunkt nicht verwendet wird, verweigert die Richtlinie jeglichen Zugriff auf den Bucket. Die `aws:SourceVpce` Bedingung gibt den Endpunkt an. Die `aws:SourceVpce` Bedingung erfordert keinen Amazon-Ressourcennamen (ARN) für die VPC Endpunktressource, sondern nur die VPC Endpunkt-ID. Weitere Informationen über die Verwendung von Bedingungen in einer Richtlinie finden Sie unter [Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln](#).

Wichtig

- Bevor Sie die folgende Beispielrichtlinie verwenden, ersetzen Sie die VPC Endpunkt-ID durch einen für Ihren Anwendungsfall geeigneten Wert. Andernfalls können Sie nicht auf Ihren Bucket zugreifen.
- Diese Richtlinie deaktiviert den Konsolenzugriff auf den angegebenen Bucket, da Konsolenanfragen nicht vom angegebenen VPC Endpunkt stammen.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::awsexamplebucket1",
                  "arn:aws:s3:::awsexamplebucket1/*"],
      "Condition": {
        "StringNotEquals": {
```

```
        "aws:SourceVpce": "vpce-1a2b3c4d"
      }
    }
  }
]
```

Beschränken Sie den Zugriff auf einen bestimmten VPC

Mithilfe der Bedingung können Sie eine Bucket-Richtlinie erstellen, die den Zugriff auf eine bestimmte VPC Kategorie einschränkt. `aws:SourceVpce` Dies ist nützlich, wenn Sie mehrere VPC Endgeräte auf demselben VPC Gerät konfiguriert haben und den Zugriff auf Ihre Amazon S3 S3-Buckets für all Ihre Endgeräte verwalten möchten. Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die Dritten den Zugriff auf `awsexamplebucket1` und ihre Objekte verweigert. VPC `vpc-111bbb22` Wenn die angegebene Option `VPC` nicht verwendet wird, verweigert die Richtlinie jeglichen Zugriff auf den Bucket. Diese Anweisung gewährt keinen Zugriff auf den Bucket. Um Zugriff zu gewähren, müssen Sie eine separate `Allow` Erklärung hinzufügen. Für den `vpc-111bbb22` Bedingungsschlüssel ist kein ARN für die VPC Ressource erforderlich, sondern nur die VPC ID.

Important

- Bevor Sie die folgende Beispielrichtlinie verwenden, ersetzen Sie die VPC ID durch einen Wert, der Ihrem Anwendungsfall entspricht. Andernfalls können Sie nicht auf Ihren Bucket zugreifen.
- Diese Richtlinie deaktiviert den Konsolenzugriff auf den angegebenen Bucket, da Konsolenanfragen nicht aus dem angegebenen VPC Bucket stammen.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909153",
  "Statement": [
    {
      "Sid": "Access-to-specific-VPC-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::awsexamplebucket1",
                  "arn:aws:s3:::awsexamplebucket1/*"],
    }
  ]
}
```

```
    "Condition": {
      "StringNotEquals": {
        "aws:SourceVpc": "vpc-111bbb22"
      }
    }
  ]
}
```

Beispiele für Amazon S3 S3-Bucket-Richtlinien

Mit Amazon-S3-Bucket-Richtlinien können Sie den Zugriff auf Objekte in Ihren Buckets sichern, sodass nur Benutzer mit den entsprechenden Berechtigungen darauf zugreifen können. Sie können sogar verhindern, dass authentifizierte Benutzer ohne die entsprechenden Berechtigungen auf Ihre Amazon-S3-Ressourcen zugreifen.

Dieser Abschnitt veranschaulicht Beispiele für typische Anwendungsfälle für Bucket-Richtlinien. Diese Beispielrichtlinien verwenden *amzn-s3-demo-bucket* als Ressourcenwert. Wenn Sie diese Richtlinien testen möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen (z. B. Ihren Bucket-Namen).

Um einer Gruppe von Objekten Berechtigungen zu gewähren oder zu verweigern, können Sie Platzhalterzeichen (*) in Amazon-Ressourcennamen (ARNs) und anderen Werten verwenden. Sie können beispielsweise den Zugriff auf Objektgruppen steuern, die mit einem gemeinsamen [Präfix](#) beginnen oder mit einer bestimmten Erweiterung enden, wie `.html` z.

Weitere Informationen zur AWS Identity and Access Management (IAM) Richtlinienprache, siehe [Richtlinien und Berechtigungen in Amazon S3](#).

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Note

Beim Testen von Berechtigungen unter Verwendung der Amazon-S3-Konsole erteilen Sie zusätzliche Berechtigungen, die die Konsole benötigt – `s3:ListAllMyBuckets`, `s3:GetBucketLocation` und `s3:ListBucket`. Ein detailliertes Beispiel für eine Richtlinie, die Berechtigungen für Benutzer erteilt und diese Berechtigungen unter Verwendung der Konsole testet, finden Sie unter [Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#).

Zu den zusätzlichen Ressourcen für die Erstellung von Bucket-Richtlinien gehören:

- Eine Liste der IAM Richtlinienaktionen, Ressourcen und Bedingungsschlüssel, die Sie bei der Erstellung einer Bucket-Richtlinie verwenden können, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.
- Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).
- Anleitungen zur Erstellung einer S3-Richtlinie finden Sie unter [Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole](#).
- Informationen zur Behebung von Fehlern mit einer Richtlinie finden Sie unter [Problembehandlung bei Zugriff verweigert \(403 Forbidden\) Fehler in Amazon S3](#).

Themen


- [Einem öffentlichen anonymen Benutzer nur Lesezugriff gewähren](#)
- [Erfordern von Verschlüsselung](#)
- [Verwaltung von Buckets mithilfe von vorgefertigten ACLs](#)
- [Verwaltung des Objektzugriffs mit Objektmarkierung](#)
- [Verwalten des Objektzugriffs mithilfe globaler Bedingungsschlüssel](#)
- [Verwaltung des Zugriffs auf der Grundlage HTTP unserer HTTPS Anfragen](#)
- [Verwalten des Benutzerzugriffs auf bestimmte Ordner](#)
- [Verwalten des Zugriffs für Zugriffsprotokolle](#)
- [Den Zugang zu einem Amazon verwalten CloudFront OAI](#)
- [Verwalten des Zugriffs für Amazon S3 Storage Lens](#)
- [Verwalten von Berechtigungen für S3 Inventory, S3 Analytics und S3-Inventory-Berichte](#)
- [Erforderlich MFA](#)
- [Benutzer daran hindern, Objekte zu löschen](#)

Einem öffentlichen anonymen Benutzer nur Lesezugriff gewähren

Sie können Ihre Richtlinieneinstellungen verwenden, um öffentlichen anonymen Benutzern Zugriff zu gewähren. Dies ist nützlich, wenn Sie Ihren Bucket als statische Website konfigurieren. Um öffentlichen anonymen Benutzern Zugriff zu gewähren, müssen Sie die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihren Bucket deaktivieren. Weitere Informationen dazu und zu

den erforderlichen Richtlinien finden Sie unter [Festlegen von Berechtigungen für den Website-Zugriff](#). Informationen zum Einrichten restriktiverer Richtlinien für denselben Zweck finden Sie unter [Wie kann ich öffentlichen Lesezugriff auf einige Objekte in meinem Amazon S3 S3-Bucket gewähren?](#) in der AWS Wissenszentrum.


Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

 **Warning**

Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#) dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugangs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#) 



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

- Block *all* public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die [Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten](#), bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Erfordern von Verschlüsselung

Sie können eine serverseitige Verschlüsselung mit verlangen AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), wie in den folgenden Beispielen gezeigt.

Erforderlich SSE — KMS für alle Objekte, die in einen Bucket geschrieben werden

Die folgende Beispielrichtlinie verlangt, dass jedes Objekt, das in den Bucket geschrieben wird, mit serverseitiger Verschlüsselung verschlüsselt wird AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS). Wenn das Objekt nicht mit SSE - verschlüsselt ist KMS, wird die Anfrage abgelehnt.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [{
    "Sid": "DenyObjectsThatAreNotSSEKMS",
    "Principal": "*",
    "Effect": "Deny",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "Null": {
        "s3:x-amz-server-side-encryption-aws-kms-key-id": "true"
      }
    }
  }]
}
```

SSE erforderlich — KMS mit einem bestimmten AWS KMS key für alle Objekte, die in einen Bucket geschrieben wurden

Die folgende Beispielrichtlinie verhindert, dass Objekte in den Bucket geschrieben werden, wenn sie nicht mit SSE - KMS unter Verwendung einer bestimmten KMS Schlüssel-ID verschlüsselt sind. Selbst wenn die Objekte mit SSE — KMS unter Verwendung einer Header- oder Bucket-Standardverschlüsselung pro Anfrage verschlüsselt sind, können die Objekte nicht in den Bucket geschrieben werden, wenn sie nicht mit dem angegebenen KMS Schlüssel verschlüsselt wurden. Achten Sie darauf, den in diesem Beispiel verwendeten KMS Schlüssel ARN durch Ihren eigenen KMS Schlüssel ARN zu ersetzen.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [{
    "Sid": "DenyObjectsThatAreNotSSEKMSWithSpecificKey",
    "Principal": "*",
```

```

"Effect": "Deny",
"Action": "s3:PutObject",
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
"Condition": {
  "ArnNotEqualsIfExists": {
    "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:kms:us-
east-2:111122223333:key/01234567-89ab-cdef-0123-456789abcdef"
  }
}
}]
}

```

Verwaltung von Buckets mithilfe von vorgefertigten ACLs

Erteilen von Berechtigungen für mehrere Konten, um Objekte hochzuladen oder Objekte öffentlich zugänglich ACLs zu machen

Die folgende Beispielrichtlinie gewährt mehreren s3:PutObject Personen die s3:PutObjectACL Berechtigungen und AWS-Konten. Außerdem verlangt die Beispielrichtlinie, dass alle Anfragen für diese Operationen die gespeicherte `public-read` [Zugriffskontrollliste \(ACL\)](#) enthalten müssen. Weitere Informationen erhalten Sie unter [Politische Maßnahmen für Amazon S3](#) und [Schlüssel für Richtlinienbedingungen für Amazon S3](#).

Warning

ACL Mit der `public-read` Option „Gespeichert“ kann jeder auf der Welt die Objekte in Ihrem Bucket einsehen. Seien Sie vorsichtig, wenn Sie anonymen Zugriff auf Ihren Amazon-S3-Bucket gewähren oder die Block-Einstellungen für den öffentlichen Zugriff deaktivieren. Wenn Sie anonymen Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, niemals anonymen Zugriff auf Ihren Amazon-S3-Bucket zu gewähren, es sei denn, Sie müssen dies ausdrücklich tun, z. B. beim [Hosting von statischen Websites](#). Informationen zum Aktivieren der Einstellungen zum Blockieren des öffentlichen Zugriffs für das Hosten statischer Websites finden Sie unter [Tutorial: Konfiguration einer statischen Website in Amazon S3](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

    "Sid": "AddPublicReadCannedAcl",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:root",
        "arn:aws:iam::444455556666:root"
      ]
    },
    "Action": [
      "s3:PutObject",
      "s3:PutObjectAcl"
    ],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "StringEquals": {
        "s3:x-amz-acl": [
          "public-read"
        ]
      }
    }
  }
}

```

Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt

Das folgende Beispiel zeigt, wie Sie ein anderes zulassen AWS-Konto um Objekte in Ihren Bucket hochzuladen und gleichzeitig sicherzustellen, dass Sie die volle Kontrolle über die hochgeladenen Objekte haben. Diese Richtlinie gewährt einen bestimmten AWS-Konto (**111122223333**) die Möglichkeit, Objekte nur hochzuladen, wenn das Konto beim Upload auch die bucket-owner-full-control ACL gescannten Objekte enthält. Die StringEquals Bedingung in der Richtlinie gibt den s3:x-amz-acl Bedingungsschlüssel an, mit dem die vordefinierte ACL Anforderung ausgedrückt werden soll. Weitere Informationen finden Sie unter [Schlüssel für Richtlinienbedingungen für Amazon S3](#).

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"PolicyForAllowUploadWithACL",
      "Effect":"Allow",

```

```
"Principal":{"AWS":"111122223333"},
"Action":"s3:PutObject",
"Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
"Condition": {
  "StringEquals": {"s3:x-amz-acl":"bucket-owner-full-control"}
}
]
}
```

Verwaltung des Objektzugriffs mit Objektmarkierung

Einem Benutzer nur das Lesen von Objekten gestatten, die einen bestimmten Tag-Schlüssel und -Wert besitzen

Die folgende Berechtigungsrichtlinie beschränkt einen Benutzer darauf, nur Objekte zu lesen, die den Tag-Schlüssel und -Wert `environment: production` haben. Diese Richtlinie verwendet den `s3:ExistingObjectTag`-Bedingungsschlüssel, um den Tag-Schlüssel und -Wert anzugeben.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Principal":{"
        "AWS":"arn:aws:iam::111122223333:role/JohnDoe"
      }},
      "Effect":"Allow",
      "Action":[
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition":{"
        "StringEquals":{"
          "s3:ExistingObjectTag/environment":"production"
        }
      }
    }
  ]
}
```

Einschränken, welche Objekt-Tag-Schlüssel Benutzer hinzufügen können

Die folgende Beispielrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen Objekt Markierungen hinzuzufügen. Die Bedingung verwendet den Bedingungsschlüssel `s3:RequestObjectTagKeys`, um die zulässigen Tag-Schlüssel zu spezifizieren, z. B. `Owner` oder `CreationDate`. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Bedingung, die mehrere Schlüsselwerte testet](#).

Die Richtlinie stellt sicher, dass jeder in der Anfrage angegebene Tag-Schlüssel ein autorisierter Tag-Schlüssel ist. Der `ForAnyValue`-Qualifizierer in der Bedingung stellt sicher, dass mindestens einer der spezifizierten Schlüssel in der Anfrage enthalten ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Einen spezifischen Tag-Schlüssel und -Wert verlangen, wenn Benutzern erlaubt wird, Objekt-Tags hinzuzufügen

Die folgende Beispielrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen

Objekt Markierungen hinzuzufügen. Die Bedingung verlangt, dass der Benutzer einen spezifischen Tag-Schlüssel (z. B. *Project*) mit dem Wert *X* angibt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:RequestObjectTag/Project": "X"
        }
      }
    }
  ]
}
```

Einem Benutzer nur das Hinzufügen mit einem bestimmten Tag-Schlüssel und -Wert erlauben

Die folgende Beispielrichtlinie gewährt einem Benutzer die Berechtigung, die `s3:PutObject`-Aktion auszuführen, sodass er Objekte einem Bucket hinzufügen kann. Die `Condition`-Anweisung schränkt die Tag-Schlüssel und -Werte jedoch ein, die für die hochgeladenen Objekte zulässig sind. In diesem Beispiel kann der Benutzer dem Bucket nur Objekte hinzufügen, die den spezifischen Tag-Schlüssel (*Department*) mit dem Wert *Finance* haben.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": [
```

```
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {
        "StringEquals": {
            "s3:RequestObjectTag/Department": "Finance"
        }
    }
}]]
}
```

Verwalten des Objektzugriffs mithilfe globaler Bedingungsschlüssel

[Globale Bedingungsschlüssel](#) sind Bedingungskontextschlüssel mit einem aws Präfix. AWS-Services kann globale Bedingungsschlüssel oder dienstspezifische Schlüssel unterstützen, die das Dienstpräfix enthalten. Sie können das Condition Element einer JSON Richtlinie verwenden, um die Schlüssel in einer Anfrage mit den Schlüsselwerten zu vergleichen, die Sie in Ihrer Richtlinie angeben.

Beschränken des Zugriffs nur auf Zugriffsprotokollbereitstellungen von Amazon S3 Server

Im folgenden Beispiel ist die Bucket-Richtlinie [aws:SourceArn](#) Der globale Bedingungsschlüssel wird verwendet, um den [Amazon-Ressourcennamen \(ARN\)](#) der Ressource zu vergleichen und eine service-to-service Anfrage mit ARN dem in der Richtlinie angegebenen zu stellen. Sie können diesen globalen `aws:SourceArn`-Bedingungsschlüssel verwenden, um zu verhindern, dass der Amazon-S3-Service bei Transaktionen zwischen Services als [verwechelter Stellvertreter](#) verwendet wird. Nur der Amazon-S3-Service darf dem Amazon-S3-Bucket Objekte hinzufügen.

Diese Bucket-Beispielrichtlinie gewährt nur dem Prinzipal des Protokollierungsservices (`logging.s3.amazonaws.com`) `s3:PutObject`-Berechtigungen.

Note

Die [NotPrincipal](#) Element kann nicht verwendet werden mit AWS-Service Prinzipale in ressourcenbasierten Amazon S3 S3-Richtlinien, wie z. B. Bucket-Richtlinien. Stattdessen empfehlen wir, den `aws:PrincipalServiceName` Bedingungsschlüssel zu verwenden, wie in der folgenden Richtlinie dargestellt.


```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutObjectS3ServerAccessLogsPolicy",
      "Principal": {
        "Service": "logging.s3.amazonaws.com"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-Logs/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111111111111"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::EXAMPLE-SOURCE-BUCKET"
        }
      }
    },
    {
      "Sid": "RestrictToS3ServerAccessLogs",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-Logs/*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:PrincipalServiceName": "logging.s3.amazonaws.com"
        }
      }
    }
  ]
}

```

Erlauben des Zugriffs nur für Ihre Organisation

Wenn Sie möchten, dass alle [IAMPrincipals](#), die auf eine Ressource zugreifen, von einem AWS-Konto in Ihrer Organisation (einschließlich der AWS Organizations Verwaltungskonto) können Sie den `aws:PrincipalOrgID` globalen Bedingungsschlüssel verwenden.

Wenn Sie diese Art von Zugriff gewähren oder einschränken möchten, definieren Sie die `aws:PrincipalOrgID`-Bedingung und legen Sie den Wert in der Bucket-Richtlinie auf Ihre

[Organisations-ID](#) fest. Die Organisations-ID wird verwendet, um den Zugriff auf den Bucket zu kontrollieren. Wenn Sie die `aws:PrincipalOrgID`-Bedingung verwenden, werden die Berechtigungen aus der Bucket-Richtlinie auch auf alle neuen Konten angewendet, die der Organisation hinzugefügt werden.

Hier ist ein Beispiel für eine ressourcenbasierte Bucket-Richtlinie, mit der Sie bestimmten IAM Prinzipalen in Ihrer Organisation direkten Zugriff auf Ihren Bucket gewähren können. Durch das Hinzufügen des globalen Bedingungsschlüssels `aws:PrincipalOrgID` zu Ihrer Bucket-Richtlinie muss sich das Prinzipal-Konto nun in Ihrer Organisation befinden, um Zugriff auf die Ressource zu erhalten. Selbst wenn Sie bei der Gewährung des Zugriffs versehentlich ein falsches Konto angeben, [aws:PrincipalOrgID Der globale Bedingungsschlüssel](#) dient als zusätzlicher Schutz. Wenn dieser globale Schlüssel in einer Richtlinie verwendet wird, verhindert diese, dass alle Prinzipale außerhalb der angegebenen Organisation auf den Amazon-S3-Bucket zugreifen können. Nur Prinzipale von Konten in der aufgelisteten Organisation erhalten Zugriff auf die Ressource.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowGetObject",
    "Principal": {
      "AWS": "*"
    },
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "StringEquals": {
        "aws:PrincipalOrgID": ["o-aa111bb222"]
      }
    }
  }]
}
```

Verwaltung des Zugriffs auf der Grundlage HTTP unserer HTTPS Anfragen

Beschränken Sie den Zugriff nur auf HTTPS Anfragen

Wenn Sie verhindern möchten, dass potenzielle Angreifer den Netzwerkverkehr manipulieren, können Sie HTTPS (TLS) verwenden, um nur verschlüsselte Verbindungen zuzulassen und gleichzeitig den Zugriff von HTTP Anfragen auf Ihren Bucket einzuschränken. Um festzustellen, ob es sich bei der Anfrage um HTTP oder handelt HTTPS, verwenden Sie

`aws:SecureTransport` globaler Bedingungsschlüssel in Ihrer S3-Bucket-Richtlinie. Der `aws:SecureTransport` Bedingungsschlüssel prüft, ob eine Anfrage mithilfe von `HTTPS` gesendet wurde. `HTTP`.

Wenn eine Anfrage zurückkehrt `true`, wurde die Anfrage durchgesendet `HTTPS`. Wenn die Anfrage zurückkehrt `false`, wurde die Anfrage durchgesendet `HTTP`. Sie können dann den Zugriff auf Ihren Bucket basierend auf dem gewünschten Anforderungsschema erlauben oder verweigern.

Im folgenden Beispiel lehnt die Bucket-Richtlinie `HTTP` Anfragen ausdrücklich ab.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "RestrictToTLSRequestsOnly",
    "Action": "s3:*",
    "Effect": "Deny",
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ],
    "Condition": {
      "Bool": {
        "aws:SecureTransport": "false"
      }
    },
    "Principal": "*"
  }]
}
```

Beschränken Sie den Zugriff auf einen bestimmten `HTTP` Referer

Angenommen, Sie haben eine Website mit dem Domainnamen `www.example.com` oder `example.com` mit Links zu Fotos und Videos, die in Ihrem Bucket gespeichert sind `amzn-s3-demo-bucket`. Standardmäßig sind alle Amazon S3 S3-Ressourcen privat, also nur AWS-Konto der die Ressourcen erstellt hat, kann auf sie zugreifen.

Wenn Sie Lesezugriff auf diese Objekte von Ihrer Website aus erlauben möchten, können Sie eine Bucket-Richtlinie hinzufügen, die die `s3:GetObject`-Berechtigung mit der Bedingung erlaubt, dass die `GET`-Anforderung von bestimmten Webseiten stammen muss. Die folgende Richtlinie schränkt Anforderungen ein, indem die Bedingung `StringLike` zusammen mit dem `aws:Referer`-Bedingungsschlüssel verwendet wird.

```
{
  "Version":"2012-10-17",
  "Id":"HTTP referer policy example",
  "Statement":[
    {
      "Sid":"Allow only GET requests originating from www.example.com and
example.com.",
      "Effect":"Allow",
      "Principal":"*",
      "Action":["s3:GetObject","s3:GetObjectVersion"],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition":{"
        "StringLike":{"aws:Referer":["http://www.example.com/*","http://example.com/
*"]}}
      }
    ]
  }
}
```

Stellen Sie sicher, dass die von Ihnen verwendeten Browser den HTTP referer Header in die Anfrage aufnehmen.

Warning

Wir empfehlen, bei der Verwendung des `aws:Referer`-Bedingungsschlüssels vorsichtig zu sein. Es ist gefährlich, einen öffentlich bekannten HTTP Referer-Header-Wert einzufügen. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:Referer`-Wert ihrer Wahl bereitstellen. Verwenden Sie es daher nicht, `aws:Referer` um zu verhindern, dass Unbefugte direkt AWS Anfragen.

Der `aws:Referer`-Bedingungsschlüssel wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [aws:Referer](#) im IAM-Benutzerhandbuch.

Verwalten des Benutzerzugriffs auf bestimmte Ordner

Erteilen des Benutzerzugriffs auf bestimmte Ordner

Angenommen, Sie versuchen, Benutzern Zugriff auf einen bestimmten Ordner zu gewähren. Wenn der IAM Benutzer und der S3-Bucket zum selben gehören AWS-Konto, dann können Sie dem

Benutzer mithilfe einer IAM Richtlinie Zugriff auf einen bestimmten Bucket-Ordner gewähren. Bei diesem Ansatz müssen Sie Ihre Bucket-Richtlinie nicht aktualisieren, um Zugriff zu gewähren. Sie können die IAM Richtlinie einer IAM Rolle hinzufügen, zu der mehrere Benutzer wechseln können.

Wenn die IAM Identität und der S3-Bucket zu unterschiedlichen gehören AWS-Konten, dann müssen Sie sowohl in der Richtlinie als auch in der IAM Bucket-Richtlinie kontoübergreifenden Zugriff gewähren. Weitere Informationen zum Gewähren von kontoübergreifendem Zugriff finden Sie unter [Bucket-Eigentümer erteilt kontoübergreifende Bucket-Berechtigungen](#).

Die folgende Bucket-Beispielrichtlinie gewährt *JohnDoe* vollen Konsolenzugriff nur auf seinen Ordner (home/*JohnDoe*/). Indem Sie einen home-Ordner erstellen und Ihren Benutzern die entsprechenden Berechtigungen gewähren, können sich mehrere Benutzer einen einzelnen Bucket teilen. Diese Richtlinie besteht aus drei Allow-Anweisungen:

- *AllowRootAndHomeListingOfCompanyBucket*: Erlaubt dem Benutzer (*JohnDoe*), Objekte auf der Stammebene des *DOC-EXAMPLE-BUCKET*-Buckets und im home-Ordner aufzulisten. Diese Anweisung erlaubt dem Benutzer außerdem, mithilfe der Konsole nach dem Präfix home/ zu suchen.
- *AllowListingOfUserFolder*: Erlaubt dem Benutzer (*JohnDoe*), alle Objekte im home/*JohnDoe*/-Ordner und entsprechenden Unterordnern aufzulisten.
- *AllowAllS3ActionsInUserFolder*: Erlaubt dem Benutzer, alle Amazon-S3-Aktionen durchzuführen, indem Read-, Write- und Delete-Berechtigungen gewährt werden. Die Berechtigungen sind auf den Home-Ordner des Bucket-Eigentümers beschränkt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRootAndHomeListingOfCompanyBucket",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/JohnDoe"
        ]
      },
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
      "Condition": {
        "StringEquals": {
```

```

        "s3:prefix": [""], "home/", "home/JohnDoe"],
        "s3:delimiter": ["/"]
    }
},
{
    "Sid": "AllowListingOfUserFolder",
    "Principal": {
        "AWS": [
            "arn:aws:iam::111122223333:user/JohnDoe"
        ]
    },
    "Action": ["s3:ListBucket"],
    "Effect": "Allow",
    "Resource": ["arn:aws:s3::DOC-EXAMPLE-BUCKET"],
    "Condition": {
        "StringLike": {
            "s3:prefix": ["home/JohnDoe/*"]
        }
    }
},
{
    "Sid": "AllowAllS3ActionsInUserFolder",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::111122223333:user/JohnDoe"
        ]
    },
    "Action": ["s3:*"],
    "Resource": ["arn:aws:s3::DOC-EXAMPLE-BUCKET/home/JohnDoe/*"]
}
]
}

```

Verwalten des Zugriffs für Zugriffsprotokolle

Gewähren des Zugriffs auf Application Load Balancer, um Zugriffsprotokolle zu aktivieren

Wenn Sie die Zugriffsprotokollierung für Ihren Application Load Balancer aktivieren, müssen Sie den Namen des S3-Buckets angeben, in dem der Load Balancer [die Protokolle speichert](#). Dem Bucket muss eine [Richtlinie angefügt](#) sein, die Elastic Load Balancing die Berechtigung zum Schreiben in den Bucket gewährt.

Im folgenden Beispiel gewährt die Bucket-Richtlinie Elastic Load Balancing (ELB) die Berechtigung, die Zugriffs-Logs in den Bucket zu schreiben:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

Note

Stellen Sie sicher, dass Sie es durch das *elb-account-id* ersetzen AWS-Konto ID für Elastic Load Balancing für Ihren AWS-Region. Eine Liste der Elastic Load Balancing Balancing-Regionen finden Sie unter [Eine Richtlinie an Ihren Amazon S3 S3-Bucket anhängen](#) im Elastic Load Balancing Balancing-Benutzerhandbuch.

Wenn Ihre AWS-Region erscheint nicht in der Liste der unterstützten Elastic Load Balancing Balancing-Regionen. Verwenden Sie die folgende Richtlinie, die dem angegebenen Log-Lieferdienst Berechtigungen gewährt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

```
}
```

Stellen Sie anschließend sicher, dass Sie Ihre [Zugriffsprotokolle von Elastic Load Balancing](#) konfigurieren, indem Sie sie aktivieren. Sie können [Ihre Bucket-Berechtigungen überprüfen](#), indem Sie eine Testdatei erstellen.

Den Zugang zu einem Amazon verwalten CloudFront OAI

Erteilen Sie einem Amazon die Erlaubnis CloudFront OAI

Die folgende Beispiel-Bucket-Richtlinie gewährt einer CloudFront Origin-Zugriffsidentität (OAI) die Berechtigung, alle Objekte in Ihrem S3-Bucket abzurufen (zu lesen). Sie können a verwenden CloudFront OAI, um Benutzern den Zugriff auf Objekte in Ihrem Bucket über Amazon S3 zu ermöglichen, CloudFront aber nicht direkt über Amazon S3. Weitere Informationen finden Sie unter [Beschränken des Zugriffs auf Amazon S3 S3-Inhalte mithilfe einer Origin-Zugriffsidentität](#) im Amazon CloudFront Developer Guide.

In der folgenden Richtlinie wird die ID OAI der Richtlinie als die ID der Richtlinie verwendet. Principal Weitere Informationen zur Verwendung von S3-Bucket-Richtlinien zur Gewährung des Zugriffs auf eine CloudFront OAI finden Sie unter [Migration von der ursprünglichen Zugriffsidentität \(OAI\) zur Ursprungszugriffskontrolle \(OAC\)](#) im Amazon CloudFront Developer Guide.

Zur Verwendung dieses Beispiels gehen Sie wie folgt vor:

- Ersetze es *EH1HDMB1FH2TC* durch OAI die ID. Um die OAI s-ID zu finden, besuche die [Origin Access Identity-Seite](#) auf der CloudFront Konsole oder verwende [ListCloudFrontOriginAccessIdentities](#) in der CloudFront API.
- Ersetzen Sie *amzn-s3-demo-bucket* durch den Namen von Ihrem Bucket.

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForCloudFrontPrivateContent",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
Identity EH1HDMB1FH2TC"
      },
      "Action": "s3:GetObject",
```



```
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

Verwalten des Zugriffs für Amazon S3 Storage Lens

Erteilen von Berechtigungen für Amazon S3 Storage Lens

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Ihr Dashboard bietet Drilldown-Optionen zur Generierung und Visualisierung von Erkenntnissen über die Organisation, den Account, AWS-Region, Speicherklasse, Bucket, Präfix oder Storage Lens auf Gruppenebene. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren.

S3 Storage Lens kann Ihre aggregierten Speichernutzungsmetriken in einen Amazon-S3-Bucket exportieren, um sie weiter zu analysieren. Der Bucket, in dem S3 Storage Lens seine Metrikexporte speichert, wird als Ziel-Bucket bezeichnet. Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket haben, wenn Sie Ihren Metrikexport in S3 Storage Lens einrichten. Weitere Informationen finden Sie unter [Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens](#).

Die folgende Bucket-Beispielrichtlinie erteilt Amazon S3 die Berechtigung, Objekte (PUT-Anforderungen) in einen Ziel-Bucket zu schreiben. Sie verwenden eine solche Bucket-Richtlinie für den Ziel-Bucket, wenn Sie einen S3-Storage-Lens-Metrik-Export einrichten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3StorageLensExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "storage-lens.s3.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/destination-prefix/StorageLens/111122223333/*"
      ]
    }
  ]
}
```

```

    ],
    "Condition": {
      "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control",
        "aws:SourceAccount": "111122223333",
        "aws:SourceArn": "arn:aws:s3:region-code:111122223333:storage-
lens/storage-lens-dashboard-configuration-id"
      }
    }
  }
]
}

```

Verwenden Sie die folgende Änderung an der vorherigen Resource-Anweisung der Bucket-Richtlinie beim Einrichten eines Metrikexports von S3 Storage Lens auf Organisationsebene.

```

"Resource": "arn:aws:s3::amzn-s3-demo-destination-bucket/destination-prefix/
StorageLens/your-organization-id/*",

```

Verwalten von Berechtigungen für S3 Inventory, S3 Analytics und S3-Inventory-Berichte

Gewähren von Berechtigungen für S3 Inventory und S3 Analytics

S3 Inventory erstellt Listen der Objekte in einem Bucket und der Speicherklassenanalyse-Export von S3 Analytics erstellt Ausgabedateien der in der Analyse verwendeten Daten. Der Bucket, dessen Objekte die Bestandserfassung auflistet, wird als Quell-Bucket bezeichnet. Der Bucket, in dem die Bestandsdatei und die Analyseexportdatei geschrieben werden, wird als Ziel-Bucket bezeichnet. Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket erstellen, wenn Sie den Bestand oder einen Analyseexport einrichten. Weitere Informationen erhalten Sie unter [Amazon S3 Inventory](#) und [Amazon S3 analytics – Speicherklassen-Analyse](#).

Das folgende Beispiel für eine Bucket-Richtlinie erteilt Amazon S3 die Berechtigung, Objekte aus dem Konto für den Quell-Bucket in den Ziel-Bucket zu schreiben (PUT-Anforderungen). Sie verwenden eine solche Bucket-Richtlinie für den Ziel-Bucket, wenn Sie einen S3-Inventory- und S3-Analytics-Export einrichten.

```

{
  "Version": "2012-10-17",

```

```

    "Statement": [
      {
        "Sid": "InventoryAndAnalyticsExamplePolicy",
        "Effect": "Allow",
        "Principal": {
          "Service": "s3.amazonaws.com"
        },
        "Action": "s3:PutObject",
        "Resource": [
          "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
        ],
        "Condition": {
          "ArnLike": {
            "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
          },
          "StringEquals": {
            "aws:SourceAccount": "111122223333",
            "s3:x-amz-acl": "bucket-owner-full-control"
          }
        }
      }
    ]
  }
}

```

Steuern Sie die Erstellung der Konfiguration des S3-Inventarberichts

[Amazon S3 Inventory](#) erstellt Listen der Objekte in einem S3-Bucket sowie die Metadaten für jedes Objekt. Die `s3:PutInventoryConfiguration` Berechtigung ermöglicht es einem Benutzer, eine Inventarkonfiguration zu erstellen, die alle standardmäßig verfügbaren Objektmetadatenfelder enthält, und den Ziel-Bucket zum Speichern des Inventars anzugeben. Ein Benutzer mit Lesezugriff auf Objekte im Ziel-Bucket kann auf alle Objektmetadatenfelder zugreifen, die im Bestandsbericht verfügbar sind. Weitere Informationen über die Metadatenfelder, die in S3 Inventory verfügbar sind, finden Sie unter [Amazon-S3-Inventory-Liste](#).

Um einen Benutzer daran zu hindern, einen S3-Inventarbericht zu konfigurieren, entziehen Sie dem Benutzer die `s3:PutInventoryConfiguration` Erlaubnis.

Einige Objektmetadatenfelder in Konfigurationen für S3-Inventarberichte sind optional, d. h. sie sind standardmäßig verfügbar, können aber eingeschränkt werden, wenn Sie einem Benutzer die `s3:PutInventoryConfiguration` entsprechende Berechtigung erteilen. Mithilfe des `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssels können Sie steuern, ob Benutzer diese optionalen Metadatenfelder in ihre Berichte aufnehmen können. Eine Liste der

optionalen Metadatenfelder, die in S3 Inventory verfügbar sind, finden Sie unter [OptionalFields](#) in der Amazon Simple Storage Service API Reference.

Um einem Benutzer die Erlaubnis zu erteilen, eine Inventarkonfiguration mit bestimmten optionalen Metadatenfeldern zu erstellen, verwenden Sie den `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssel, um die Bedingungen in Ihrer Bucket-Richtlinie zu verfeinern.

Die folgende Beispielrichtlinie gewährt einem Benutzer (*Ana*) unter bestimmten Bedingungen die Berechtigung, eine Inventarkonfiguration zu erstellen. Die `ForAllValues:StringEquals` Bedingung in der Richtlinie verwendet den `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssel, um die beiden zulässigen optionalen Metadatenfelder anzugeben, nämlich `Size` und `StorageClass`. Wenn *Ana* sie also eine Inventarkonfiguration erstellt, sind die einzigen optionalen Metadatenfelder, die sie einbeziehen kann, `Size` und `StorageClass`.

```
{
  "Id": "InventoryConfigPolicy",
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowInventoryCreationConditionally",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:user/Ana"
    },
    "Action":
      "s3:PutInventoryConfiguration",
    "Resource":
      "arn:aws:s3::DOC-EXAMPLE-SOURCE-BUCKET",
    "Condition": {
      "ForAllValues:StringEquals": {
        "s3:InventoryAccessibleOptionalFields": [
          "Size",
          "StorageClass"
        ]
      }
    }
  ]
}
```

Um einen Benutzer daran zu hindern, einen S3-Inventarbericht zu konfigurieren, der bestimmte optionale Metadatenfelder enthält, fügen Sie der Bucket-Richtlinie für den Quell-Bucket eine


ausdrückliche Deny Anweisung hinzu. Die folgende Beispiel-Bucket-Richtlinie verbietet *Ana* dem Benutzer, eine Inventarkonfiguration im Quell-Bucket zu erstellen, die *DOC-EXAMPLE-SOURCE-BUCKET* die optionalen Felder `ObjectAccessControlList` oder `ObjectOwner` Metadatenfelder enthält. Der Benutzer *Ana* kann weiterhin eine Inventarkonfiguration mit anderen optionalen Metadatenfeldern erstellen.

```
{
  "Id": "InventoryConfigSomeFields",
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowInventoryCreation",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::<111122223333>:user/Ana"
    },
    "Action": "s3:PutInventoryConfiguration",
    "Resource":
      "arn:aws:s3::",

  },
  {
    "Sid": "DenyCertainInventoryFieldCreation",
    "Effect": "Deny",
    "Principal": {
      "AWS": "arn:aws:iam::<111122223333>:user/Ana"
    },
    "Action": "s3:PutInventoryConfiguration",
    "Resource":
      "arn:aws:s3::",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "s3:InventoryAccessibleOptionalFields": [
          "ObjectOwner",
          "ObjectAccessControlList"
        ]
      }
    }
  }
]
```

 Note

Die Verwendung des `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssels in Bucket-Richtlinien hat keinen Einfluss auf die Bereitstellung von Inventarberichten, die auf den vorhandenen Inventarkonfigurationen basieren.

 Important

Wir empfehlen die Verwendung `ForAllValues` mit einem `Allow` Effekt oder `ForAnyValue` mit einem `Deny` Effekt, wie in den vorherigen Beispielen gezeigt.

Verwenden Sie die Option weder `ForAllValues` mit `Deny` Wirkung noch `ForAnyValue` mit `Allow` Wirkung, da diese Kombinationen zu restriktiv sein und das Löschen der Inventarkonfiguration blockieren können.

Weitere Informationen zu den Operatoren `ForAllValues` und `ForAnyValue` Bedingungssatzoperatoren finden Sie im Benutzerhandbuch unter [Mehrwertige Kontexttasten](#). IAM

Erforderlich MFA

Amazon S3 unterstützt den MFA -geschützten API Zugriff, eine Funktion, die eine Multi-Faktor-Authentifizierung (MFA) für den Zugriff auf Ihre Amazon S3 S3-Ressourcen erzwingen kann. Die Multi-Faktor-Authentifizierung bietet eine zusätzliche Sicherheitsstufe, die Sie auf Ihre AWS Umgebung. MFA ist eine Sicherheitsfunktion, bei der Benutzer den physischen Besitz eines MFA Geräts nachweisen müssen, indem sie einen gültigen MFA Code angeben. Weitere Informationen finden Sie unter [AWS Multi-Faktor-Authentifizierung](#). Sie können MFA für alle Anfragen den Zugriff auf Ihre Amazon S3 S3-Ressourcen verlangen.

Um die MFA Anforderung durchzusetzen, verwenden Sie den `aws:MultiFactorAuthAge` Bedingungsschlüssel in einer Bucket-Richtlinie. IAM Benutzer können auf Amazon S3 S3-Ressourcen zugreifen, indem sie temporäre Anmeldeinformationen verwenden, die von der AWS Security Token Service (AWS STS). Sie geben den MFA Code zum Zeitpunkt der AWS STS Anfrage.

Wenn Amazon S3 eine Anforderung mit Multi-Factor Authentication empfängt, liefert der `aws:MultiFactorAuthAge`-Bedingungsschlüssel einen numerischen Wert, der angibt, wie lange (in Sekunden) es her ist, dass die temporären Anmeldeinformationen erstellt wurden. Wenn die in der Anfrage angegebenen temporären Anmeldeinformationen nicht mithilfe eines MFA Geräts erstellt

wurden, ist dieser Schlüsselwert Null (fehlt). In einer Bucket-Richtlinie können Sie eine Bedingung hinzufügen, um diesen Wert zu überprüfen, wie im folgenden Beispiel gezeigt.

Diese Beispielrichtlinie verweigert jegliche Amazon S3 S3-Operation auf dem */steuerdokumente* Ordner im *amzn-s3-demo-bucket* Bucket, wenn die Anfrage nicht mithilfe von authentifiziert wurde. MFA Weitere Informationen MFA dazu finden Sie unter [Verwenden der Multi-Faktor-Authentifizierung \(\) MFA in AWS](#) im IAM-Benutzerhandbuch.

```
{
  "Version": "2012-10-17",
  "Id": "123",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
      "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
    }
  ]
}
```

Die Null Bedingung im Condition Block wird dahingehend ausgewertet, true ob der Wert des `aws:MultiFactorAuthAge` Bedingungsschlüssels Null ist, was darauf hinweist, dass die temporären Sicherheitsanmeldedaten in der Anforderung ohne ein MFA Gerät erstellt wurden.

Die folgende Bucket-Richtlinie ist eine Erweiterung der vorhergehenden Bucket-Richtlinie. Die folgende Richtlinie umfasst zwei Richtlinienerklärungen. Eine Anweisung gewährt jedem die `s3:GetObject`-Berechtigung für einen Bucket (*amzn-s3-demo-bucket*). Eine weitere Anweisung schränkt den Zugriff auf den *amzn-s3-demo-bucket/taxdocuments* Ordner im Bucket weiter ein, indem sie Folgendes verlangtMFA.

```
{
  "Version": "2012-10-17",
  "Id": "123",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Deny",
      "Principal": "*",
```

```

    "Action": "s3:*",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
    "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
  },
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": "*",
    "Action": ["s3:GetObject"],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
  }
]
}

```

Optional können Sie eine numerische Bedingung verwenden, um die Gültigkeitsdauer des `aws:MultiFactorAuthAge`-Schlüssels zu beschränken. Die Dauer, die Sie für den `aws:MultiFactorAuthAge`-Schlüssel angeben, ist unabhängig von der Lebensdauer der temporären Sicherheitsanmeldeinformationen, die für die Authentifizierung der Anforderung verwendet wurden.

Die folgende Bucket-Richtlinie verlangt beispielsweise nicht nur eine MFA Authentifizierung, sondern überprüft auch, wie lange es her ist, dass die temporäre Sitzung erstellt wurde. Die Richtlinie verweigert jede Operation, wenn der `aws:MultiFactorAuthAge`-Schlüsselwert angibt, dass die temporäre Sitzung vor mehr als einer Stunde erstellt wurde (3600 Sekunden).

```

{
  "Version": "2012-10-17",
  "Id": "123",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
      "Condition": {"Null": {"aws:MultiFactorAuthAge": true }}
    },
    {
      "Sid": "",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",

```



```

    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
    "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }}
  },
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": "*",
    "Action": ["s3:GetObject"],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
  }
]
}

```

Benutzer daran hindern, Objekte zu löschen

Standardmäßig besitzen Benutzer keine Berechtigungen. Beim Erstellen von Richtlinien können Sie Benutzern jedoch Berechtigungen gewähren, die Sie nicht gewähren wollten. Um solche Berechtigungslücken zu vermeiden, können Sie eine strengere Zugriffsrichtlinie verfassen, indem Sie eine ausdrückliche Verweigerung hinzufügen.

Um Benutzer oder Konten explizit am Löschen von Objekten zu hindern, müssen Sie einer Bucket-Richtlinie die folgenden Aktionen hinzufügen: `s3:DeleteObjects`, `s3:DeleteObjectVersion`, und Berechtigungen. `s3:PutLifecycleConfiguration`. Alle drei Aktionen sind erforderlich, da Sie Objekte löschen können, indem Sie entweder die `DELETE Object` API Operationen explizit aufrufen oder ihren Lebenszyklus konfigurieren (siehe [Verwaltung des Lebenszyklus von Objekten](#)), sodass Amazon S3 die Objekte entfernen kann, wenn ihre Lebensdauer abläuft.

Im folgenden Richtlinienbeispiel verweigern Sie dem Benutzer ausdrücklich `DELETE Object` Berechtigungen *MaryMajor*. Eine ausdrückliche Deny Erklärung hat immer Vorrang vor allen anderen erteilten Genehmigungen.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
      },
      "Action": [

```

```

        "s3:GetObjectVersion",
        "s3:GetBucketAcl"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
    ]
},
{
    "Sid": "statement2",
    "Effect": "Deny",
    "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
    },
    "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutLifecycleConfiguration"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
    ]
}
]
}

```

Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln

Sie können die Sprache der Zugriffsrichtlinie verwenden, um Bedingungen anzugeben, wenn Sie Berechtigungen erteilen. Sie können das optionale Condition Element oder den Condition Block verwenden, um Bedingungen für das Inkrafttreten einer Richtlinie festzulegen.

Richtlinien, die Amazon-S3-Bedingungsschlüssel für Objekt- und Bucket-Vorgänge verwenden, finden Sie in den folgenden Beispielen. Weitere Informationen über Bedingungsschlüssel finden Sie unter [Schlüssel für Richtlinienbedingungen für Amazon S3](#). Eine vollständige Liste der Amazon S3-Aktionen, Bedingungsschlüssel und Ressourcen, die Sie in Richtlinien angeben können, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Beispiele: Amazon S3 S3-Bedingungsschlüssel für Objektoperationen

Die folgenden Beispiele zeigen, wie Sie Amazon S3 S3-spezifische Bedingungsschlüssel für Objektoperationen verwenden können. Eine vollständige Liste der Amazon S3-Aktionen, Bedingungsschlüssel und Ressourcen, die Sie in Richtlinien angeben können, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Einige der Beispielrichtlinien zeigen, wie Sie Bedingungsschlüssel mit [PUTObjektoperationen](#) verwenden können. PUTObjektoperationen ermöglichen Zugriffskontrolllisten (ACL), d. h. Header, die Sie verwenden können, um bestimmte Berechtigungen zu erteilenACL. Mithilfe dieser Bedingungsschlüssel können Sie eine Bedingung festlegen, die bestimmte Zugriffsberechtigungen erfordert, wenn der Benutzer ein Objekt hochlädt. Sie können im Rahmen des Vorgangs auch ACL Berechtigungen auf der Grundlage von Berechtigungen gewähren. PutObjectAcl Weitere Informationen finden Sie [PutObjectAcl](#) in der Amazon S3 API S3-Referenz zu Amazon Simple Storage Service. Weitere Informationen zu finden ACLs Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Themen

- [Beispiel 1: Erteilung einer s3:PutObject Berechtigung, die erfordert, dass Objekte mit serverseitiger Verschlüsselung gespeichert werden](#)
- [Beispiel 2: Erteilen der s3:PutObject Erlaubnis zum Kopieren von Objekten mit einer Einschränkung der Kopierquelle](#)
- [Beispiel 3: Zugriff auf eine bestimmte Version eines Objekts gewähren](#)
- [Beispiel 4: Erteilen von Berechtigungen auf der Grundlage von Objekt-Tags](#)
- [Beispiel 5: Beschränkung des Zugriffs durch AWS-Konto ID des Bucket-Besitzers](#)
- [Beispiel 6: Eine TLS Mindestversion ist erforderlich](#)
- [Beispiel 7: Bestimmte Hauptpersonen aus einer Deny Anweisung ausschließen](#)

Beispiel 1: Erteilung einer **s3:PutObject** Berechtigung, die erfordert, dass Objekte mit serverseitiger Verschlüsselung gespeichert werden

Angenommen, Konto A besitzt einen Bucket. Der Kontoadministrator möchte Jane, einer Benutzerin in Konto A, die Erlaubnis zum Hochladen von Objekten unter der Bedingung gewähren, dass Jane immer serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) anfordert.

Der Konto-A-Administrator kann diese Anforderung mithilfe des `s3:x-amz-server-side-encryption` Bedingungsschlüssels wie in der Abbildung dargestellt spezifizieren. Das Schlüssel-Wert-Paar im folgenden Condition Block gibt den `s3:x-amz-server-side-encryption` Bedingungsschlüssel und SSE -S3 (AES256) als Verschlüsselungstyp an:

```
"Condition": {
  "StringNotEquals": {
    "s3:x-amz-server-side-encryption": "AES256"
  }
}
```

Beim Testen dieser Berechtigung mit dem AWS CLI, müssen Sie die erforderliche Verschlüsselung mithilfe des `--server-side-encryption` Parameters hinzufügen, wie im folgenden Beispiel gezeigt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key HappyFace.jpg --body c:\HappyFace.jpg --server-side-encryption "AES256" --profile AccountAdmin
```

Beispiel 2: Erteilen der **s3:PutObject** Erlaubnis zum Kopieren von Objekten mit einer Einschränkung der Kopierquelle

Wenn Sie in einer PUT Objektanforderung ein Quellobjekt angeben, handelt es sich bei der Anforderung um einen Kopiervorgang (siehe [CopyObject](#)). Dementsprechend kann der Bucket-Besitzer einem Benutzer die Erlaubnis erteilen, Objekte mit Einschränkungen in Bezug auf die Quelle zu kopieren, zum Beispiel:

- Erlaubt das Kopieren von Objekten nur aus dem angegebenen Quell-Bucket (z. B. *amzn-s3-demo-source-bucket*).
- Erlaubt das Kopieren von Objekten aus dem angegebenen Quell-Bucket und nur von Objekten, deren Schlüsselnamenpräfix mit einem bestimmten Präfix beginnt, z. B. *öffentlich/* (zum Beispiel *amzn-s3-demo-source-bucket/public/*).
- Erlaubt das Kopieren nur eines bestimmten Objekts aus dem Quell-Bucket (z. B. *amzn-s3-demo-source-bucket/example.jpg*).

Die folgende Bucket-Richtlinie gewährt einem Benutzer (*Dave*) die `s3:PutObject` Berechtigung. Diese Richtlinie erlaubt es ihm, Objekte nur unter der Bedingung zu kopieren, dass die Anfrage den `s3:x-amz-copy-source` Header enthält und dass der Header-Wert das */amzn-s3-demo-*

*source-bucket/public/** Schlüsselnamenpräfix angibt. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "cross-account permission to user in your own account",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::amzn-s3-demo-source-bucket/*"
    },
    {
      "Sid": "Deny your user permission to upload object if copy source is not /
bucket/prefix",
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::amzn-s3-demo-source-bucket/*",
      "Condition": {
        "StringNotLike": {
          "s3:x-amz-copy-source": "amzn-s3-demo-source-bucket/public/*"
        }
      }
    }
  ]
}
```

Testen Sie die Richtlinie mit dem AWS CLI

Sie können die Genehmigung testen, indem Sie den AWS CLI `copy-object` Befehl. Sie geben die Quelle an, indem Sie den `--copy-source` Parameter hinzufügen. Das von Ihnen angegebene Schlüsselnamenpräfix muss mit dem in der Richtlinie zulässigen Präfix übereinstimmen. Sie müssen die *Dave* Benutzeranmeldeinformationen mithilfe des `--profile` Parameters angeben. Weitere Informationen zur Einrichtung des AWS CLI, finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#). Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api copy-object --bucket amzn-s3-demo-source-bucket --key HappyFace.jpg
--copy-source amzn-s3-demo-source-bucket/public/PublicHappyFace1.jpg --
profile AccountADave
```

Erteilen der Berechtigung, nur ein bestimmtes Objekt zu kopieren

Die vorherige Richtlinie verwendet die Bedingung `StringNotLike`. Um die Erlaubnis zu erteilen, nur ein bestimmtes Objekt zu kopieren, müssen Sie die Bedingung von `StringNotLike` bis ändern `StringNotEquals` und dann den genauen Objektschlüssel angeben, wie im folgenden Beispiel gezeigt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
"Condition": {
  "StringNotEquals": {
    "s3:x-amz-copy-source": "amzn-s3-demo-source-bucket/public/
PublicHappyFace1.jpg"
  }
}
```

Beispiel 3: Zugriff auf eine bestimmte Version eines Objekts gewähren

Angenommen, Konto A besitzt einen Bucket mit aktivierter Versionierung. Der Bucket beinhaltet mehrere Versionen des Objekts *HappyFace.jpg*. Der Administrator des Kontos A möchte dem Benutzer nun die *Dave* Erlaubnis erteilen, nur eine bestimmte Version des Objekts abzurufen. Der Kontoadministrator kann dies erreichen, indem er dem Benutzer *Dave* die `s3:GetObjectVersion` Berechtigung unter bestimmten Bedingungen erteilt, wie im folgenden Beispiel gezeigt. Das Schlüssel-Wert-Paar im Block `Condition` gibt den Bedingungsschlüssel `s3:VersionId` an. In diesem Fall *Dave* muss die genaue Versions-ID des Objekts bekannt sein, um das Objekt aus dem angegebenen Bucket mit aktivierter Versionierung abzurufen. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Weitere Informationen finden Sie unter [GetObject](#) in der Amazon Simple Storage Service API Reference.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
```

```

    "Principal": {
      "AWS": "arn:aws:iam::123456789012:user/Dave"
    },
    "Action": "s3:GetObjectVersion",
    "Resource": "arn:aws:s3::amzn-s3-demo-bucket/HappyFace.jpg"
  },
  {
    "Sid": "statement2",
    "Effect": "Deny",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:user/Dave"
    },
    "Action": "s3:GetObjectVersion",
    "Resource": "arn:aws:s3::amzn-s3-demo-bucket/HappyFace.jpg",
    "Condition": {
      "StringNotEquals": {
        "s3:VersionId": "AaaHbAQitwiL_h47_44lR02DDfLLB05e"
      }
    }
  }
]
}

```

Testen Sie die Richtlinie mit AWS CLI

Sie können die Berechtigungen in dieser Richtlinie testen, indem Sie den AWS CLI `get-object` Befehl mit dem `--version-id` Parameter zur Identifizierung der spezifischen Objektversion, die abgerufen werden soll. Der Befehl ruft die angegebene Version des Objekts ab und speichert sie in der *OutputFile.jpg* Datei.

```
aws s3api get-object --bucket amzn-s3-demo-bucket --key HappyFace.jpg OutputFile.jpg --version-id AaaHbAQitwiL_h47_44lR02DDfLLB05e --profile AccountADave
```

Beispiel 4: Erteilen von Berechtigungen auf der Grundlage von Objekt-Tags

Beispiele für die Verwendung von Bedingungsschlüsseln für Objekt-Tagging bei Amazon S3 S3-Vorgängen finden Sie unter [Markierungs- und Zugriffskontrollrichtlinien](#).

Beispiel 5: Beschränkung des Zugriffs durch AWS-Konto ID des Bucket-Besitzers

Sie können entweder den Bedingungsschlüssel `aws:ResourceAccount` oder den `s3:ResourceAccount` Bedingungsschlüssel verwenden, um IAM oder virtuelle private Cloud

(VPC) -Endpunktrichtlinien zu schreiben, die den Benutzer-, Rollen- oder Anwendungszugriff auf die Amazon S3 S3-Buckets einschränken, die einer bestimmten Person gehören AWS-Konto ID. Sie können diese Bedingungsschlüssel verwenden, um zu verhindern, dass Ihre VPC Clients auf Buckets zugreifen, die Sie nicht besitzen.

Beachten Sie jedoch, dass einige AWS Dienste sind abhängig vom Zugriff auf AWS verwaltete Buckets. Daher kann sich die Verwendung der `s3:ResourceAccount` Taste `aws:ResourceAccount` oder in Ihrer IAM Richtlinie auch auf den Zugriff auf diese Ressourcen auswirken. Weitere Informationen finden Sie in den folgenden Ressourcen:

- [Beschränken Sie den Zugriff auf Buckets in einem bestimmten Bereich AWS-Konto](#) in der AWS PrivateLink Leitfaden
- [Beschränken Sie den Zugriff auf Buckets, die Amazon im Amazon ECR Guide ECR verwendet](#)
- [Stellen Sie den erforderlichen Zugriff auf Systems Manager bereit für AWS verwaltete Amazon S3 S3-Buckets](#) in der AWS Systems Manager Leitfaden

Weitere Informationen zu den Schlüsseln `aws:ResourceAccount` und den `s3:ResourceAccount` Bedingungsschlüsseln sowie Beispiele für deren Verwendung finden Sie unter [Beschränken des Zugriffs auf Amazon S3 S3-Buckets, die bestimmten Personen gehören AWS-Konten](#) in der AWS Speicher-Blog.

Beispiel 6: Eine TLS Mindestversion ist erforderlich

Sie können den `s3:TlsVersion` Bedingungsschlüssel verwenden IAM, um Richtlinien für virtuelle private Cloud-Endpunkte (VPCE) oder Buckets zu schreiben, die den Benutzer- oder Anwendungszugriff auf Amazon S3 S3-Buckets je nach der vom Client verwendeten TLS Version einschränken. Sie können diesen Bedingungsschlüssel verwenden, um Richtlinien zu schreiben, für die eine TLS Mindestversion erforderlich ist.

Example

Die folgende Beispiel-Bucket-Richtlinie lehnt PutObject Anfragen von Clients ab, deren TLS Version älter als 1.2 ist, z. B. 1.1 oder 1.0. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
```



```
"Statement": [
  {
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket1",
      "arn:aws:s3:::amzn-s3-demo-bucket1/*"
    ],
    "Condition": {
      "NumericLessThan": {
        "s3:TlsVersion": 1.2
      }
    }
  }
]
```

Example

Die folgende Beispiel-Bucket-Richtlinie erlaubt PutObject Anfragen von Clients mit einer späteren TLS Version als 1.1, zum Beispiel 1.2, 1.3 oder höher:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ],
      "Condition": {
        "NumericGreaterThan": {
          "s3:TlsVersion": 1.1
        }
      }
    }
  ]
}
```

}

Beispiel 7: Bestimmte Hauptpersonen aus einer **Deny** Anweisung ausschließen

Die folgende Bucket-Richtlinie verweigert den `s3:GetObject` Zugriff auf die `amzn-s3-demo-bucket`, außer für Principals mit der Kontonummer `123456789012`. Um diese Beispielrichtlinie zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessFromPrincipalNotInSpecificAccount",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": [
            "123456789012"
          ]
        }
      }
    }
  ]
}
```

Beispiele: Amazon S3 S3-Bedingungsschlüssel für Bucket-Operationen

Die folgenden Beispielrichtlinien zeigen, wie Sie Amazon S3 S3-spezifische Bedingungsschlüssel für Bucket-Operationen verwenden können.

Themen

- [Beispiel 1: Erteilen einer s3:GetObject Erlaubnis mit einer Bedingung für eine IP-Adresse](#)
- [Beispiel 2: Abrufen einer Liste von Objekten in einem Bucket mit einem bestimmten Präfix](#)

- [Beispiel 3: Festlegen der maximalen Anzahl von Schlüsseln](#)

Beispiel 1: Erteilen einer **s3:GetObject** Erlaubnis mit einer Bedingung für eine IP-Adresse

Sie können authentifizierten Benutzern die Erlaubnis erteilen, die `s3:GetObject` Aktion zu verwenden, wenn die Anfrage von einem bestimmten IP-Adressbereich stammt (z. B. `192.0.2.*`), es sei denn, es handelt sich um eine IP-Adresse, die Sie ausschließen möchten (z. B.).

`192.0.2.188` Im Condition Block `IpAddress` und `NotIpAddress` befinden sich Bedingungen, und jeder Bedingung wird ein Schlüssel-Wert-Paar zur Auswertung zugewiesen. Beide Schlüssel-Wert-Paare in diesem Beispiel verwenden den `aws:SourceIp` AWS breiter Schlüssel. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Note

Die im Condition Block angegebenen `NotIpAddress` Schlüsselwerte `IpAddress` und verwenden die CIDR Notation, wie in RFC 4632 beschrieben. Weitere Informationen finden Sie unter <http://www.rfc-editor.org/rfc/rfc4632.txt>.

```
{
  "Version": "2012-10-17",
  "Id": "S3PolicyId1",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "192.0.2.0/24"
        },
        "NotIpAddress": {
          "aws:SourceIp": "192.0.2.188/32"
        }
      }
    }
  ]
}
```

}

Sie können auch andere verwenden AWS-weite Bedingungsschlüssel in Amazon S3 S3-Richtlinien. Sie können beispielsweise die Bedingungsschlüssel `aws:SourceVpce` und die `aws:SourceVpc` Bedingungsschlüssel in Bucket-Richtlinien für VPC Endgeräte angeben. Spezifische Beispiele finden Sie unter [Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien](#).

Note

Für einige AWS Bei globalen Bedingungsschlüsseln werden nur bestimmte Ressourcentypen unterstützt. Prüfen Sie daher, ob Amazon S3 den globalen Bedingungsschlüssel und den Ressourcentyp unterstützt, den Sie verwenden möchten, oder ob Sie stattdessen einen Amazon S3 S3-spezifischen Bedingungsschlüssel verwenden müssen. Eine vollständige Liste der unterstützten Ressourcentypen und Bedingungsschlüssel für Amazon S3 finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Beispiel 2: Abrufen einer Liste von Objekten in einem Bucket mit einem bestimmten Präfix

Sie können den `s3:prefix` Bedingungsschlüssel verwenden, um die Antwort des [ListObjectsV2](#) API-Operation zur Eingabe von Schlüsselnamen mit einem bestimmten Präfix. Wenn Sie der Bucket-Besitzer sind, können Sie diesen Bedingungsschlüssel verwenden, um einen Benutzer darauf zu beschränken, den Inhalt eines bestimmten Präfixes im Bucket aufzulisten. Der `s3:prefix` Bedingungsschlüssel ist nützlich, wenn die Objekte im Bucket nach Schlüsselnamenpräfixen organisiert sind.

Die Amazon-S3-Konsole verwendet Schlüsselnamenpräfixe zum Anzeigen von Ordnerkonzepten. Nur die Konsole unterstützt das Konzept von Ordnern; Amazon S3 API unterstützt nur Buckets und Objekte. Zum Beispiel, wenn Sie zwei Objekte mit den Schlüsselnamen haben `public/object1.jpg` and `public/object2.jpg`, die Konsole zeigt die Objekte unter `öffentlich` folder. In Amazon S3 API sind dies Objekte mit Präfixen, keine Objekte in Ordnern. Weitere Informationen zur Verwendung von Präfixen und Trennzeichen zum Filtern von Zugriffsberechtigungen finden Sie unter [Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#).

Im folgenden Szenario sind der Bucket-Besitzer und das übergeordnete Konto, zu dem der Benutzer gehört, identisch. Der Bucket-Besitzer kann also entweder eine Bucket-Richtlinie oder eine Benutzerrichtlinie verwenden, um Zugriff zu gewähren. Weitere Informationen zu anderen Bedingungsschlüsseln, die Sie für den ListObjectsV2 API Vorgang verwenden können, finden Sie unter [ListObjectsV2](#).

Note

Wenn für den Bucket die Versionsverwaltung aktiviert ist, müssen Sie, um die Objekte im Bucket aufzulisten, die `s3:ListBucketVersions` Berechtigung in den folgenden Richtlinien statt der `s3:ListBucket` Berechtigung erteilen. Die `s3:ListBucketVersions` Berechtigung unterstützt auch den `s3:prefix` Bedingungsschlüssel.

Richtlinie für Benutzer:

Die folgende Benutzerrichtlinie gewährt die `s3:ListBucket` Berechtigung (siehe [ListObjectsV2](#)) mit einer `Condition` Anweisung, nach der der Benutzer in der Anfrage ein Präfix mit dem Wert angeben muss *projects*. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {
        "StringEquals": {
          "s3:prefix": "projects"
        }
      }
    },
    {
      "Sid": "statement2",
      "Effect": "Deny",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {
```

```

        "StringNotEquals" : {
            "s3:prefix": "projects"
        }
    }
}
]
}

```

Die Condition Anweisung beschränkt den Benutzer darauf, nur Objektschlüssel aufzulisten, die das *projects* Präfix haben. Die hinzugefügte explizite Deny Anweisung verbietet dem Benutzer, Schlüssel mit einem anderen Präfix aufzulisten, unabhängig davon, welche anderen Berechtigungen der Benutzer möglicherweise hat. Es ist beispielsweise möglich, dass der Benutzer die Erlaubnis erhält, Objektschlüssel ohne Einschränkungen aufzulisten, entweder durch Aktualisierungen der vorherigen Benutzerrichtlinie oder durch eine Bucket-Richtlinie. Da explizite Deny Anweisungen immer Allow Anweisungen überschreiben, wird die Anfrage verweigert, wenn der Benutzer versucht, andere Schlüssel als die aufzulisten, die das *projects* Präfix haben.

Bucket-Richtlinie

Wenn Sie das Principal Element zur obigen Benutzerrichtlinie hinzufügen und den Benutzer identifizieren, haben Sie jetzt eine Bucket-Richtlinie, wie im folgenden Beispiel gezeigt. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {
        "StringEquals": {
          "s3:prefix": "projects"
        }
      }
    }
  ],
}

```

```
{
  "Sid": "statement2",
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
  },
  "Action": "s3:ListBucket",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
  "Condition": {
    "StringNotEquals": {
      "s3:prefix": "projects"
    }
  }
}
```

Testen Sie die Richtlinie mit dem AWS CLI

Sie können die Richtlinie wie folgt testen `list-object` AWS CLI Befehl. Im Befehl geben Sie die Benutzeranmeldeinformationen mithilfe des `--profile` Parameters an. Weitere Informationen zum Einrichten und Verwenden des AWS CLI, finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

```
aws s3api list-objects --bucket amzn-s3-demo-bucket --prefix projects --
profile AccountA
```

Beispiel 3: Festlegen der maximalen Anzahl von Schlüsseln

Sie können den `s3:max-keys` Bedingungsschlüssel verwenden, um die maximale Anzahl von Schlüsseln festzulegen, die ein Anforderer in einem [ListObjectsV2](#) oder [ListObjectVersions](#) request. Standardmäßig geben diese API Operationen bis zu 1.000 Schlüssel zurück. Eine Liste der numerischen Bedingungsoperatoren, die Sie zusammen mit `s3:max-keys` zugehörigen Beispielen verwenden können, finden Sie unter [Numerische Bedingungsoperatoren](#) im IAM Benutzerhandbuch.

Identitätsbasierte Richtlinien für Amazon S3

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Amazon S3 S3-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mit dem ausführen AWS Management Console, AWS Command Line Interface (AWS CLI), oder AWS API. Um Benutzern die Erlaubnis zu erteilen,

Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

Informationen zum Erstellen einer IAM identitätsbasierten Richtlinie anhand dieser JSON Beispieldokumente finden Sie unter [IAM Richtlinien erstellen](#) im IAM Benutzerhandbuch.

Einzelheiten zu den von Amazon S3 definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon S3](#)
- [Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon S3 S3-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Diese Aktionen können Kosten für Sie verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Fangen Sie an mit AWS verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Um zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie AWS verwaltete Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar AWS-Konto. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie Folgendes definieren AWS vom Kunden verwaltete Richtlinien, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS verwaltete Richtlinien](#) oder [AWS verwaltete Richtlinien für Jobfunktionen](#) im IAM Benutzerhandbuch.
- Berechtigungen mit den geringsten Rechten anwenden — Wenn Sie Berechtigungen mit IAM Richtlinien festlegen, gewähren Sie nur die Berechtigungen, die für die Ausführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen

unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung IAM zum Anwenden von Berechtigungen finden Sie [IAMim Benutzerhandbuch unter Richtlinien und Berechtigungen](#). IAM

- Verwenden Sie Bedingungen in IAM Richtlinien, um den Zugriff weiter einzuschränken — Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen einzuschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um anzugeben, dass alle Anfragen mit gesendet werden müssenSSL. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese über eine bestimmte AWS-Service, wie beispielsweise AWS CloudFormation. Weitere Informationen finden Sie unter [IAMJSONRichtlinienelemente: Zustand](#) im IAMBenutzerhandbuch.
- Verwenden Sie IAM Access Analyzer, um Ihre IAM Richtlinien zu validieren, um sichere und funktionale Berechtigungen zu gewährleisten. IAM Access Analyzer validiert neue und bestehende Richtlinien, sodass die Richtlinien der IAM Richtlinienprache (JSON) und den IAM bewährten Methoden entsprechen. IAMAccess Analyzer bietet mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen, um Sie bei der Erstellung sicherer und funktionaler Richtlinien zu unterstützen. Weitere Informationen finden Sie unter [IAMAccess Analyzer-Richtlinienvvalidierung](#) im IAMBenutzerhandbuch.
- Multi-Faktor-Authentifizierung erforderlich (MFA) — Wenn Sie ein Szenario haben, das IAM Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, schalten Sie MFA für zusätzliche Sicherheit ein. Wenn Sie festlegen möchten, MFA wann API Operationen aufgerufen werden, fügen Sie MFA Bedingungen zu Ihren Richtlinien hinzu. Weitere Informationen finden Sie unter [Konfiguration des MFA -geschützten API Zugriffs](#) im IAMBenutzerhandbuch.

Weitere Informationen zu bewährten Methoden finden Sie unter [Bewährte Sicherheitsmethoden IAM im IAM](#) Benutzerhandbuch. IAM

Beispiele für identitätsbasierte Richtlinien für Amazon S3

Dieser Abschnitt enthält mehrere Beispiele AWS Identity and Access Management (IAM) identitätsbasierte Richtlinien zur Steuerung des Zugriffs auf Amazon S3. Beispiele für Bucket-Richtlinien (ressourcenbasierte Richtlinien) finden Sie unter. [Bucket-Richtlinien für Amazon S3](#) Informationen zur Sprache der IAM Richtlinien finden Sie unter. [Richtlinien und Berechtigungen in Amazon S3](#)

Die folgenden Beispielrichtlinien funktionieren, wenn Sie sie programmgesteuert verwenden. Um sie mit der Amazon-S3-Konsole verwenden zu können, müssen Sie aber zusätzliche Berechtigungen gewähren, die für die Konsole erforderlich sind. Informationen zur Verwendung von Richtlinien wie

dieser mit der Amazon-S3-Konsole finden Sie unter [Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#).

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Themen

- [Einem IAM Benutzer Zugriff auf einen Ihrer Buckets gewähren](#)
- [Jedem IAM Benutzer Zugriff auf einen Ordner in einem Bucket gewähren](#)
- [Einer Gruppe erlauben, einen freigegebenen Ordner in Amazon S3 zu haben](#)
- [Erteilen der Erlaubnis für alle Benutzer, Objekte in einem Teil des Buckets zu lesen](#)
- [Erteilen der Erlaubnis für einen Partner, Dateien in einem bestimmten Bereich des Buckets abzulegen](#)
- [Beschränken des Zugriffs auf Amazon S3 S3-Buckets innerhalb eines bestimmten AWS-Konto](#)
- [Beschränken Sie den Zugriff auf Amazon S3 S3-Buckets innerhalb Ihrer Organisationseinheit](#)
- [Beschränken des Zugriffs auf Amazon-S3-Buckets innerhalb Ihrer Organisation](#)
- [Erteilung der Erlaubnis zum Abrufen der PublicAccessBlock Konfiguration für ein AWS-Konto](#)
- [Beschränken der Bucket-Erstellung auf eine Region](#)

Einem IAM Benutzer Zugriff auf einen Ihrer Buckets gewähren

In diesem Beispiel möchten Sie einem IAM Benutzer in Ihrem AWS-Konto Zugriff auf einen Ihrer Buckets, *amzn-s3-demo-bucket1*, und ermöglichen es dem Benutzer, Objekte hinzuzufügen, zu aktualisieren und zu löschen.

Zusätzlich zum Erteilen der Berechtigungen `s3:PutObject`, `s3:GetObject` und `s3:DeleteObject` für den Benutzer, gewährt die Richtlinie die Berechtigungen `s3:ListAllMyBuckets`, `s3:GetBucketLocation` und `s3:ListBucket`. Dies sind die zusätzlichen Berechtigungen, die von der Konsole benötigt werden. Außerdem sind die Aktionen `s3:PutObjectAcl` und `s3:GetObjectAcl` erforderlich, um Objekte in der Konsole kopieren, ausschneiden und einfügen zu können. Ein detailliertes Beispiel für eine Richtlinie, die Berechtigungen für Benutzer erteilt und sie unter Verwendung der Konsole testet, finden Sie unter [Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#).

```
{
```

```
"Version":"2012-10-17",
"Statement":[
  {
    "Effect":"Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  },
  {
    "Effect":"Allow",
    "Action":["s3:ListBucket", "s3:GetBucketLocation"],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1"
  },
  {
    "Effect":"Allow",
    "Action":[
      "s3:PutObject",
      "s3:PutObjectAcl",
      "s3:GetObject",
      "s3:GetObjectAcl",
      "s3:DeleteObject"
    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
  }
]
```

Jedem IAM Benutzer Zugriff auf einen Ordner in einem Bucket gewähren

In diesem Beispiel möchten Sie, dass zwei IAM Benutzer, Mary und Carlos, Zugriff auf Ihren Bucket haben. *amzn-s3-demo-bucket1*, sodass sie Objekte hinzufügen, aktualisieren und löschen können. Allerdings möchten Sie den Zugriff beider Benutzer auf ein einzelnes Präfix (Ordner) im Bucket einschränken. Sie können Ordner mit Namen erstellen, die ihren Benutzernamen entsprechen.

```
amzn-s3-demo-bucket1
  Mary/
  Carlos/
```

Um jedem Benutzer nur den Zugriff auf den eigenen Ordner zu gewähren, können Sie für jeden Benutzer eine Richtlinie schreiben und ihnen einzeln zuweisen. Sie können beispielsweise die folgende Richtlinie der Benutzerin Mary zuweisen, um ihr spezifische Amazon-S3-Berechtigungen für den Ordner *amzn-s3-demo-bucket1/Mary* zu gewähren.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource":["arn:aws:s3:::amzn-s3-demo-bucket1/Mary/*"]
    }
  ]
}
```

Dann können Sie dem Benutzer Carlos eine ähnliche Richtlinie zuweisen, indem Sie den Ordner *Carlos* im Wert Resource angeben.

Statt Richtlinien einzelnen Benutzern zuzuweisen, können Sie auch eine einzelne Richtlinie mit einer Richtlinienvariable schreiben und dann die Richtlinie einer Gruppe zuweisen. Sie müssen zuerst eine Gruppe erstellen und die Benutzer Mary und Carlos in die Gruppe aufnehmen. Die folgende Beispielrichtlinie erlaubt eine Reihe von Amazon-S3-Berechtigungen für den Ordner *amzn-s3-demo-bucket1/\${aws:username}*. Bei der Auswertung der Richtlinie wird die Richtlinienvariable *\${aws:username}* durch den Benutzernamen des Anforderers ersetzt. Wenn Mary beispielsweise eine Anforderung zum Anlegen eines Objekts sendet, ist die Operation nur zulässig, wenn Mary das Objekt in den Ordner *amzn-s3-demo-bucket1/Mary* hochlädt.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],

```

```

    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
  }
]
}

```

Note

Bei der Verwendung von Richtlinienvariablen müssen Sie explizit die Version 2012-10-17 in der Richtlinie angeben. Die Standardversion der IAM Richtliniensprache, 17.10.2008, unterstützt keine Richtlinienvariablen.

Wenn Sie die vorherige Richtlinie in der Amazon-S3-Konsole testen möchten, erfordert die Konsole zusätzliche Berechtigungen, wie in der folgenden Richtlinie gezeigt. Weitere Informationen darüber, wie die Konsole diese Berechtigungen verwendet, finden Sie unter [Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3::*"
    },
    {
      "Sid": "AllowRootLevelListingOfTheBucket",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
      "Condition": {
        "StringEquals": {
          "s3:prefix": [""], "s3:delimiter": ["/"]
        }
      }
    },
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",

```

```

    "Action": "s3:ListBucket",
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
    "Condition":{ "StringLike":{"s3:prefix":["${aws:username}/*"]} }
  },
  {
    "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
    "Effect":"Allow",
    "Action":[
      "s3:PutObject",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "s3:DeleteObject",
      "s3:DeleteObjectVersion"
    ],
    "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
  }
]
}

```

Note

In der 2012-10-17-Version der Richtlinie beginnen Richtlinienvariablen mit \$. Diese Syntaxänderung kann möglicherweise einen Konflikt verursachen, wenn Ihr Objektschlüssel (Objektnamen) ein \$-Zeichen enthält.

Damit dieser Konflikt vermieden wird, geben Sie das Zeichen \$ mithilfe von ``${}`` an. Um beispielsweise einen Objektschlüssel `my$file` in eine Richtlinie aufzunehmen, geben Sie das Zeichen mit `my`${}`file` an.

IAMBennennamen sind zwar benutzerfreundliche, für Menschen lesbare Kennungen, müssen aber nicht global eindeutig sein. Wenn beispielsweise der Benutzer Carlos die Organisation verlässt und ein anderer Carlos hinzukommt, könnte der neue Carlos auf die Informationen des vorherigen Carlos zugreifen.

Anstatt Benutzernamen zu verwenden, könnten Sie Ordner auf der Grundlage von Benutzern erstellen. IAM IDs Jede IAM Benutzer-ID ist einzigartig. In diesem Fall müssen Sie die vorherige Richtlinie ändern, und die Richtlinienvariable `${aws:userid}` verwenden. Weitere Informationen zu Benutzerkennungen finden Sie unter [IAMIdentifikatoren](#) im IAMBennutzerhandbuch.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/home/${aws:userid}/*"
    }
  ]
}
```

Erlauben Sie IAM Nichtbenutzern (Benutzern mobiler Apps) den Zugriff auf Ordner in einem Bucket

Angenommen, Sie möchten eine mobile App entwickeln, ein Spiel, in dem Benutzerdaten in einem S3-Bucket gespeichert werden. Sie möchten für jeden App-Benutzer einen Ordner in Ihrem Bucket erstellen. Sie möchten auch den Zugriff jedes Benutzers auf seinen eigenen Ordner einschränken. Sie können jedoch keine Ordner erstellen, bevor jemand Ihre App heruntergeladen und das Spiel gespielt hat, da Sie dessen Benutzer-ID nicht haben.

In diesem Fall können Sie von den Nutzern verlangen, dass Sie sich in Ihrer App über einen öffentlichen Identitätsanbieter anmelden, wie z. B. Login with Amazon, Facebook oder Google. Nachdem sich Benutzer über einen dieser Anbieter bei Ihrer App angemeldet haben, verfügen sie über eine Benutzer-ID, mit der Sie zur Laufzeit benutzerspezifische Ordner erstellen können.

Sie können dann den Web-Identitätsverbund verwenden in AWS Security Token Service um Informationen vom Identitätsanbieter in Ihre App zu integrieren und temporäre Sicherheitsanmeldedaten für jeden Benutzer zu erhalten. Anschließend können Sie IAM Richtlinien erstellen, die es der App ermöglichen, auf Ihren Bucket zuzugreifen und beispielsweise benutzerspezifische Ordner zu erstellen und Daten hochzuladen. Weitere Informationen zum Web Identity Federation finden Sie unter [About Web Identity Federation](#) im IAMBenutzerhandbuch.

Einer Gruppe erlauben, einen freigegebenen Ordner in Amazon S3 zu haben

Durch Anfügen der folgenden Richtlinie an die Gruppe erhalten alle Benutzer der Gruppe Zugriff auf den folgenden Ordner in Amazon S3: *amzn-s3-demo-bucket1*/share/marketing.

Gruppenmitglieder dürfen nur auf die spezifischen Amazon-S3-Berechtigungen zugreifen, die in der Richtlinie gegeben sind, und nur für Objekte im angegebenen Ordner.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/share/marketing/*"
    }
  ]
}
```

Erteilen der Erlaubnis für alle Benutzer, Objekte in einem Teil des Buckets zu lesen

In diesem Beispiel erstellen Sie eine Gruppe mit dem Namen *AllUsers*, die alle IAM Benutzer enthält, deren Eigentümer AWS-Konto. Anschließend fügen Sie eine Richtlinie hinzu, die der Gruppe Zugriff auf `GetObject` und `GetObjectVersion` gewährt, jedoch nur für Objekte im *amzn-s3-demo-bucket1/readonly* Ordner.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/readonly/*"
    }
  ]
}
```


Erteilen der Erlaubnis für einen Partner, Dateien in einem bestimmten Bereich des Buckets abzulegen

In diesem Beispiel erstellen Sie eine Gruppe namens *AnyCompany*, die eine Partnerfirma darstellt. Sie erstellen einen IAM Benutzer für die bestimmte Person oder Anwendung im Partnerunternehmen, die Zugriff benötigt, und fügen den Benutzer dann der Gruppe hinzu.

Sie fügen dann eine Richtlinie hinzu, die der Gruppe den PutObject-Zugriff auf den folgenden Ordner im Bucket erteilt:

amzn-s3-demo-bucket1/uploads/anycompany

Sie möchten verhindern, dass die *AnyCompany* Gruppe etwas anderes mit dem Bucket macht, also fügen Sie eine Erklärung hinzu, die ausdrücklich die Erlaubnis zu allen Amazon S3 S3-Aktionen verweigert, mit Ausnahme PutObject von Amazon S3 S3-Ressourcen in der AWS-Konto.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
    },
    {
      "Effect": "Deny",
      "Action": "s3:*",
      "NotResource": "arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
    }
  ]
}
```

Beschränken des Zugriffs auf Amazon S3 S3-Buckets innerhalb eines bestimmten AWS-Konto

Wenn Sie sicherstellen möchten, dass Ihre Amazon S3 S3-Prinzipale nur auf die Ressourcen zugreifen, die sich in einem vertrauenswürdigen AWS-Konto, können Sie den Zugriff einschränken. Diese [identitätsbasierte IAM Richtlinie](#) verwendet beispielsweise einen Deny Effekt, um den Zugriff auf Amazon S3 S3-Aktionen zu blockieren, es sei denn, die Amazon S3 S3-Ressource, auf die zugegriffen wird, befindet sich im Konto. **222222222222** Um zu verhindern, dass ein IAM Principal in einem AWS-Konto Fügen Sie beim Zugriff auf Amazon S3 S3-Objekte außerhalb des Kontos die folgende IAM Richtlinie bei:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "222222222222"
          ]
        }
      }
    }
  ]
}
```

Note

Diese Richtlinie ersetzt nicht Ihre bestehenden IAM Zugriffskontrollen, da sie keinen Zugriff gewährt. Stattdessen dient diese Richtlinie als zusätzliche Schutzmaßnahme für Ihre anderen IAM Berechtigungen, unabhängig davon, welche Berechtigungen im Rahmen anderer IAM Richtlinien gewährt wurden.

Stellen Sie sicher, dass Sie die Konto-ID **222222222222** in der Richtlinie durch Ihre eigene ersetzen AWS-Konto. Um eine Richtlinie auf mehrere Konten anzuwenden und gleichzeitig diese Einschränkung beizubehalten, ersetzen Sie die Konto-ID durch den `aws:PrincipalAccount` Bedingungsschlüssel. Diese Bedingung erfordert, dass sich der Prinzipal und die Ressource in demselben Konto befinden müssen.

Beschränken Sie den Zugriff auf Amazon S3 S3-Buckets innerhalb Ihrer Organisationseinheit

Wenn Sie eine [Organisationseinheit \(OU\)](#) eingerichtet haben in AWS Organizations, vielleicht möchten Sie den Zugriff auf den Amazon S3 S3-Bucket auf einen bestimmten Teil Ihrer Organisation beschränken. In diesem Beispiel wird der `aws:ResourceOrgPaths`-Schlüssel verwendet, um den

Amazon-S3-Bucket-Zugriff auf eine OU in Ihrer Organisation einzuschränken. In diesem Beispiel lautet die [OU-ID](#) *ou-acroot-exampleou*. Stellen Sie sicher, dass Sie diesen Wert in Ihrer eigenen Richtlinie durch Ihre eigene Organisationseinheit ersetzenIDs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowS3AccessOutsideMyBoundary",
      "Effect": "Allow",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringNotLike": {
          "aws:ResourceOrgPaths": [
            "o-acorg/r-acroot/ou-acroot-exampleou/"
          ]
        }
      }
    }
  ]
}
```

Note

Diese Richtlinie gewährt keinen Zugriff. Stattdessen dient diese Richtlinie als Backstop für Ihre anderen IAM Berechtigungen und verhindert, dass Ihre Prinzipale außerhalb einer von der OU definierten Grenze auf Amazon S3 S3-Objekte zugreifen.

Die Richtlinie verweigert den Zugriff auf Amazon-S3-Aktionen, es sei denn, das Amazon-S3-Objekt, auf das zugegriffen wird, befindet sich in der OU *ou-acroot-exampleou* Ihrer Organisation. Die [IAMRichtlinienbedingung](#) `aws:ResourceOrgPaths`, dass ein mehrwertiger Bedingungsschlüssel einen der aufgelisteten OU-Pfade enthält. Die Richtlinie verwendet den `ForAllValues:StringNotLike` Operator, um die Werte von mit den aufgelisteten Werten `aws:ResourceOrgPaths` zu vergleichen, OUs ohne dass die Groß- und Kleinschreibung beachtet wird.

Beschränken des Zugriffs auf Amazon-S3-Buckets innerhalb Ihrer Organisation

Um den Zugriff auf Amazon S3 S3-Objekte innerhalb Ihrer Organisation einzuschränken, fügen Sie dem Stamm der Organisation eine IAM Richtlinie hinzu, die auf alle Konten in Ihrer Organisation angewendet wird. Verwenden Sie eine [Service-Control-Richtlinie \(SCP\)](#), um von Ihren IAM Prinzipalen die Einhaltung dieser Regel zu verlangen. Wenn Sie sich für eine entscheidenSCP, stellen Sie sicher, dass Sie sie gründlich [testen](#), SCP bevor Sie die Richtlinie dem Stammverzeichnis der Organisation zuordnen.

In der folgenden Beispielrichtlinie wird der Zugriff auf Amazon S3 S3-Aktionen verweigert, es sei denn, das Amazon S3 S3-Objekt, auf das zugegriffen wird, befindet sich in derselben Organisation wie der IAM Principal, der darauf zugreift:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "arn:aws:s3:::*/**",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
        }
      }
    }
  ]
}
```

Note

Diese Richtlinie gewährt keinen Zugriff. Stattdessen dient diese Richtlinie als Backstop für Ihre anderen IAM Berechtigungen und verhindert, dass Ihre Prinzipale auf Amazon S3 S3-Objekte außerhalb Ihrer Organisation zugreifen. Diese Richtlinie gilt auch für Amazon-S3-Ressourcen, die nach Inkrafttreten der Richtlinie erstellt werden.

Die [IAM Richtlinienbedingung](#) in diesem Beispiel erfordert `aws:PrincipalOrgID`, dass `aws:ResourceOrgID` und einander entsprechen. Bei dieser Anforderung müssen sich der Prinzipal, der die Anforderung stellt, und die Ressource, auf die zugegriffen wird, in derselben Organisation befinden.

Erteilung der Erlaubnis zum Abrufen der `PublicAccessBlock` Konfiguration für ein AWS-Konto

Das folgende Beispiel für eine identitätsbasierte Richtlinie erteilt einem Benutzer die `s3:GetAccountPublicAccessBlock` entsprechende Berechtigung. Für diese Berechtigungen legen Sie den Wert `Resource` auf `"*"` fest. Hinweise zur Ressource ARNs finden Sie unter.

[Richtlinienressourcen für Amazon S3](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Action": [
        "s3:GetAccountPublicAccessBlock"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Beschränken der Bucket-Erstellung auf eine Region

Nehmen wir an, dass ein AWS-Konto Administrator möchte seinem Benutzer (Dave) die Erlaubnis erteilen, einen Bucket nur in der Region Südamerika (São Paulo) zu erstellen. Der Kontoadministrator kann die folgende Benutzerrichtlinie anfügen, welche die Berechtigung `s3:CreateBucket` mit der angegebenen Bedingung gewährt. Das Schlüssel-Wert-Paar im Block `Condition` gibt den Schlüssel `s3:LocationConstraint` und die Region `sa-east-1` als seinen Wert an.

Note

In diesem Beispiel erteilt der Bucket-Eigentümer einem seiner Benutzer die Berechtigung, daher kann entweder eine Bucket-Richtlinie oder eine Benutzerrichtlinie verwendet werden. Dieses Beispiel zeigt eine Benutzerrichtlinie.

Eine Liste der Amazon S3 S3-Regionen finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"statement1",
      "Effect":"Allow",
      "Action": "s3:CreateBucket",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringLike": {
          "s3:LocationConstraint": "sa-east-1"
        }
      }
    }
  ]
}
```

Hinzufügen einer expliziten Zugriffsverweigerung

Die vorangegangene Richtlinie hindert den Benutzer an der Erstellung eines Buckets in einer anderen Region als sa-east-1. Einige andere Richtlinien gewähren diesem Benutzer möglicherweise jedoch die Berechtigung, Buckets in einer anderen Region zu erstellen. Wenn der Benutzer beispielsweise zu einer Gruppe gehört, ist der Gruppe möglicherweise eine Richtlinie angefügt, die alle Benutzer der Gruppe zum Erstellen von Buckets in einer anderen Region berechtigt. Um sicherzustellen, dass der Benutzer in keiner anderen Region die Erlaubnis erhält, Buckets zu erstellen, können Sie der obigen Richtlinie eine ausdrückliche Ablehnungsklausel hinzufügen.

Die Anweisung Deny verwendet die Bedingung StringNotLike. Das bedeutet, dass eine Anforderung zum Erstellen eines Buckets abgelehnt wird, wenn die Standorteinschränkung nicht

sa-east-1 ist. Die ausdrückliche Ablehnung erlaubt es dem Benutzer nicht, einen Bucket in einer anderen Region zu erstellen, unabhängig davon, welche anderen Berechtigungen der Benutzer erhält. Die folgende Richtlinie beinhaltet eine ausdrückliche Ablehnungsklausel.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"statement1",
      "Effect":"Allow",
      "Action": "s3:CreateBucket",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringLike": {
          "s3:LocationConstraint": "sa-east-1"
        }
      }
    },
    {
      "Sid":"statement2",
      "Effect":"Deny",
      "Action": "s3:CreateBucket",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringNotLike": {
          "s3:LocationConstraint": "sa-east-1"
        }
      }
    }
  ]
}
```

Testen Sie die Richtlinie mit dem AWS CLI

Sie können die Richtlinie wie folgt testen `create-bucket` AWS CLI Befehl. In diesem Beispiel wird die Datei `bucketconfig.txt` verwendet, um die Standortbeschränkung anzugeben. Beachten Sie den Windows Dateipfad. Sie müssen den Bucket-Namen und -Pfad entsprechend aktualisieren. Sie müssen die Anmeldeinformationen des Benutzers bereitstellen, indem Sie den Parameter `--profile` hinzufügen. Weitere Informationen zur Einrichtung und Verwendung von AWS CLI, finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

```
aws s3api create-bucket --bucket examplebucket --profile AccountADave --create-bucket-configuration file://c:/Users/someUser/bucketconfig.txt
```

Die Datei `bucketconfig.txt` gibt die Konfiguration wie folgt an.

```
{"LocationConstraint": "sa-east-1"}
```

Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien

Diese schrittweise Anleitung erklärt, wie Benutzerberechtigungen in Amazon S3 funktionieren. In diesem Beispiel erstellen Sie einen Bucket mit Ordnern. Anschließend erstellen Sie AWS Identity and Access Management IAM Benutzer in Ihrem AWS-Konto und gewähren Sie diesen Benutzern inkrementelle Berechtigungen für Ihren Amazon S3 S3-Bucket und die darin enthaltenen Ordner.

Themen

- [Grundlagen zu Buckets und Ordnern](#)
- [Walkthrough-Übersicht](#)
- [Vorbereitung auf den Walkthrough](#)
- [Schritt 1: Erstellen eines Buckets](#)
- [Schritt 2: IAM Benutzer und eine Gruppe erstellen](#)
- [Schritt 3: Stellen Sie sicher, dass IAM Benutzer keine Berechtigungen haben](#)
- [Schritt 4: Erteilen von Berechtigungen auf Gruppenebene](#)
- [Schritt 5: Erteilen Sie IAM der Benutzerin Alice spezifische Berechtigungen](#)
- [Schritt 6: Erteilen Sie dem IAM Benutzer Bob bestimmte Berechtigungen](#)
- [Schritt 7: Absichern des Ordners „Private“](#)
- [Schritt 8: Bereinigen](#)
- [Zugehörige Ressourcen](#)

Grundlagen zu Buckets und Ordnern

Das Amazon-S3-Datenmodell ist eine flache Struktur: Sie erstellen einen Bucket und der Bucket speichert Objekte. Es gibt keine Hierarchie für Unter-Buckets oder Unterordner. Sie können aber eine Ordnerhierarchie nachbilden. Tools, wie die Amazon-S3-Konsole, können eine Ansicht dieser logischen Ordner und Unterordner in Ihrem Bucket präsentieren.

Die Konsole zeigt, dass sich in einem Bucket mit dem Namen `companybucket` die drei Ordner `Private`, `Development` und `Finance` und das Objekt `s3-dg.pdf` befinden. Die Konsole verwendet die Objektnamen (Schlüssel), um eine logische Hierarchie mit Ordnern und Unterordnern zu erzeugen. Betrachten Sie die folgenden Beispiele:

- Wenn Sie den Ordner `Development` anlegen, erstellt die Konsole ein Objekt mit dem Schlüssel `Development/`. Beachten Sie den als Trennzeichen dienenden abschließenden Schrägstrich (`/`).
- Wenn Sie ein Objekt mit dem Namen `Projects1.xls` in den Ordner `Development` hochladen, lädt die Konsole das Objekt hoch und weist ihm den Schlüssel `Development/Projects1.xls` zu.

Im Schlüssel ist `Development` das [Präfix](#) und `/` ist das Trennzeichen. Amazon S3 API unterstützt Präfixe und Trennzeichen in seinem Betrieb. Beispielsweise können Sie eine Liste aller Objekte in einem Bucket mit einem bestimmten Präfix und Separator erhalten. Wenn Sie in der Konsole den Ordner `Development` öffnen, listet die Konsole die Objekte im betreffenden Ordner auf. Im folgenden Beispiel enthält der Ordner `Development` ein Objekt.

Wenn die Konsole den Ordner `Development` im Bucket `companybucket` auflistet, sendet sie eine Anforderung an Amazon S3, in der das Präfix `Development` und das Trennzeichen `/` in der Anforderung angegeben werden. Die Antwort der Konsole sieht genauso aus, wie eine Ordnerliste im Dateisystem Ihres Computers. Das vorherige Beispiel zeigt, dass der Bucket `companybucket` ein Objekt mit dem Schlüssel `Development/Projects1.xls` enthält.

Die Konsole verwendet Objektschlüssel, um eine logische Hierarchie abzuleiten. Amazon S3 hat keine physische Hierarchie. Amazon S3 hat nur Buckets, die Objekte in einer flachen Dateistruktur enthalten. Wenn Sie Objekte mit Amazon S3 erstellenAPI, können Sie Objektschlüssel verwenden, die eine logische Hierarchie implizieren. Wenn Sie eine logische Hierarchie von Objekten erstellen, können Sie den Zugriff auf einzelne Ordner verwalten, wie in dieser exemplarischen Anleitung veranschaulicht.

Bevor Sie beginnen, müssen Sie mit dem Konzept des Bucket-Inhalts auf Stammebene vertraut sein. Angenommen, Ihr Bucket `companybucket` enthält die folgenden Objekte:

- `Private/privDoc1.txt`
- `Private/privDoc2.zip`
- `Development/project1.xls`
- `Development/project2.xls`

- `Finance/Tax2011/document1.pdf`
- `Finance/Tax2011/document2.pdf`
- `s3-dg.pdf`

Diese Objektschlüssel erzeugen eine logische Hierarchie mit `Private`, `Development` und `Finance` als Stammebenen-Ordner und `s3-dg.pdf` als Stammebenen-Objekt. Wenn Sie den Bucket-Namen in der Amazon-S3-Konsole auswählen, erscheinen die Elemente der Stammebene. Die Konsole zeigt die Präfixe der obersten Ebene (`Private/`, `Development/` und `Finance/`) als Stammebenen-Ordner. Der Objektschlüssel `s3-dg.pdf` hat kein Präfix und erscheint daher als Stammebenen-Element.

Walkthrough-Übersicht

In dieser Anleitung erstellen Sie einen Bucket mit drei Ordnern (`Private`, `Development` und `Finance`).

Sie haben zwei User, Alice und Bob. Sie möchten, dass Alice nur auf den Ordner `Development` und Bob nur auf den Ordner `Finance` zugreift. Sie möchten, dass der Inhalt des Ordners `Private` privat bleibt. In der exemplarischen Vorgehensweise verwalten Sie den Zugriff, indem Sie IAM Benutzer erstellen (im Beispiel werden die Benutzernamen Alice und Bob verwendet) und ihnen die erforderlichen Berechtigungen erteilen.

IAM unterstützt auch das Erstellen von Benutzergruppen und das Erteilen von Berechtigungen auf Gruppenebene, die für alle Benutzer in der Gruppe gelten. Dies hilft Ihnen bei der Verwaltung der Berechtigungen. In dieser Übung benötigen sowohl Alice als auch Bob einige gemeinsame Berechtigungen. Sie erstellen also eine Gruppe mit dem Namen `Consultants` und fügen dann Alice und Bob der Gruppe hinzu. Sie erteilen zunächst Berechtigungen, indem Sie der Gruppe eine Gruppenrichtlinie zuweisen. Fügen Sie dann benutzerspezifische Berechtigungen durch Zuweisen von Richtlinien zu bestimmten Benutzern hinzu.

Note

In der exemplarischen Vorgehensweise werden `companybucket` als Bucket-Name Alice und Bob als IAM Benutzer und `Consultants` als Gruppenname verwendet. Da Bucket-Namen in Amazon S3 global eindeutig sein müssen, müssen Sie den Bucket-Namen durch einen von Ihnen erstellten Namen ersetzen.

Vorbereitung auf den Walkthrough

In diesem Beispiel verwenden Sie AWS-Konto Anmeldeinformationen, um IAM Benutzer zu erstellen. Zu Beginn haben diese Benutzer keine Berechtigungen. Sie gewähren diesen Benutzern nach und nach Berechtigungen, damit sie spezifische Amazon-S3-Aktionen ausführen können. Um diese Berechtigungen zu testen, melden Sie sich mit den Anmeldeinformationen eines jeden Benutzers bei der Konsole an. Wenn Sie inkrementell Berechtigungen gewähren als AWS-Konto Besitzer- und Testberechtigungen als IAM Benutzer müssen Sie sich jedes Mal mit unterschiedlichen Anmeldeinformationen an- und abmelden. Sie können diesen Test auch mit einem Browser durchführen, das Verfahren schreitet aber schneller fort, wenn Sie zwei verschiedene Browser verwenden. Verwenden Sie einen Browser, um eine Verbindung mit dem AWS Management Console mit deinem AWS-Konto Anmeldeinformationen und einen anderen Browser, um eine Verbindung mit den IAM Benutzeranmeldedaten herzustellen.

Um sich bei der anzumelden AWS Management Console mit deinem AWS-Konto Anmeldeinformationen, gehe zu <https://console.aws.amazon.com/>. Ein IAM Benutzer kann sich nicht mit demselben Link anmelden. Ein IAM Benutzer muss eine Anmeldeseite verwenden, auf der IAM - aktiviert ist. Als Kontoinhaber können Sie diesen Link Ihren Benutzern bereitstellen.

Weitere Informationen zu finden Sie IAM unter [AWS Management Console Anmeldeseite](#) im IAMBenutzerhandbuch.

Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich IAM Dashboard aus.
3. Beachten Sie den Link zur Anmeldung URL unter IAM Benutzern:.. Sie geben IAM Benutzern diesen Link, damit sie sich mit ihrem IAM Benutzernamen und Passwort an der Konsole anmelden können.

Schritt 1: Erstellen eines Buckets

In diesem Schritt melden Sie sich bei der Amazon S3 S3-Konsole an mit Ihrem AWS-Konto Anmeldeinformationen, erstellen Sie einen Bucket, fügen Sie dem Bucket Ordner hinzu und laden Sie in jeden Ordner ein oder zwei Beispieldokumente hoch.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Erstellen Sie einen Bucket.

step-by-step Anweisungen finden Sie unter [Erstellen eines Buckets](#).

3. Laden Sie ein Dokument in den Bucket hoch.

Diese Übung geht davon aus, dass sich das Dokument `s3-dg.pdf` auf der Stammebene dieses Buckets befindet. Wenn Sie andere Dokumente hochladen, ändern Sie ihren Dateinamen in `s3-dg.pdf`.

4. Fügen Sie drei Ordner mit dem Namen Private, Finance and Development zum Bucket hinzu.

step-by-step Anweisungen zum Erstellen eines Ordners finden Sie unter [Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern](#) > im Amazon Simple Storage Service-Benutzerhandbuch.

5. Laden Sie ein oder zwei Dokumente in jeden Ordner hoch.

Für diese Übung wird angenommen, dass Sie einige Dokumente in jeden Ordner hochgeladen haben, sodass der Bucket Objekte mit den folgenden Schlüsseln enthält:


- `Private/privDoc1.txt`
- `Private/privDoc2.zip`
- `Development/project1.xls`
- `Development/project2.xls`
- `Finance/Tax2011/document1.pdf`
- `Finance/Tax2011/document2.pdf`
- `s3-dg.pdf`

step-by-step Anweisungen finden Sie unter [Objekte hochladen](#).

Schritt 2: IAM Benutzer und eine Gruppe erstellen

Verwenden Sie nun die [IAMKonsole](#), um zwei IAM Benutzer, Alice und Bob, zu Ihrem hinzuzufügen AWS-Konto. step-by-step Anweisungen finden Sie unter [Einen IAM Benutzer erstellen in Ihrem AWS-Konto](#) im IAM-Benutzerhandbuch.

Erstellen Sie auch eine administrative Gruppe mit dem Namen `Consultants`. Fügen Sie dann beide Benutzer zur Gruppe hinzu. step-by-step Anweisungen finden Sie unter [IAMBenutzergruppen erstellen](#).

 Warning

Wenn Sie die Benutzer und die Gruppe erstellen, fügen Sie keine Richtlinien an, die diesen Benutzern Berechtigungen erteilen. Anfänglich haben diese Benutzer keine Berechtigungen. In den folgenden Abschnitten gewähren Sie nach und nach Berechtigungen. Zunächst müssen Sie sicherstellen, dass Sie diesen IAM Benutzern Passwörter zugewiesen haben. Sie verwenden diese Benutzer-Anmeldeinformationen zum Testen der Amazon-S3-Aktionen und zum Überprüfen, ob die Berechtigungen wie erwartet funktionieren.

step-by-step Anweisungen zum Erstellen eines neuen IAM Benutzers finden Sie unter [Einen IAM Benutzer erstellen in Ihrem AWS-Konto](#) im IAM-Benutzerhandbuch. Wenn Sie die Benutzer für diese exemplarische Vorgehensweise erstellen, wählen Sie AWS Management Console auf den [programmatischen Zugriff zugreifen und ihn löschen](#).

step-by-step Anweisungen zum Erstellen einer administrativen Gruppe finden Sie unter [Erstellen Ihres ersten IAM Administratorbenutzers und Ihrer ersten Administratorgruppe](#) im IAMBenutzerhandbuch.

Schritt 3: Stellen Sie sicher, dass IAM Benutzer keine Berechtigungen haben

Wenn Sie zwei Browser verwenden, können Sie sich jetzt mit dem zweiten Browser mit einem der IAM Benutzeranmeldedaten an der Konsole anmelden.

1. Melden Sie sich mithilfe des IAM Benutzeranmelde-links (siehe [Um Benutzern einen Anmelde-link zur Verfügung zu stellen IAM](#)) bei AWS Management Console mit einem der IAM Benutzeranmeldedaten.
2. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

Überprüfen Sie die Konsolenmeldung, die Ihnen mitteilt, dass der Zugriff verweigert wurde.

Nun können Sie damit beginnen, den Benutzern schrittweise Berechtigungen zu erteilen. Sie weisen zunächst eine Gruppenrichtlinie zu, die beiden Benutzern die erforderlichen Berechtigungen gewährt.

Schritt 4: Erteilen von Berechtigungen auf Gruppenebene

Sie möchten den Benutzern Folgendes ermöglichen:

- Auflisten aller Buckets, die dem übergeordneten Konto gehören. Um das zu tun, müssen Bob und Alice die Berechtigung für die Aktion `s3:ListAllMyBuckets` besitzen.
- Auflisten aller Elemente, Ordner und Objekte auf Stammebene im Bucket `companybucket`. Um das zu tun, müssen Bob und Alice die Berechtigung für die Aktion `s3:ListBucket` im Bucket `companybucket` besitzen.

Zuerst erstellen Sie eine Richtlinie, die diese Berechtigungen gewährt, und dann weisen Sie sie der Gruppe `Consultants` zu.

Schritt 4.1: Erteilen der Berechtigung zum Auflisten aller Buckets

In diesem Schritt erstellen Sie eine verwaltete Richtlinie, die den Benutzern die Mindestberechtigungen für die Auflistung aller Buckets des übergeordneten Kontos erteilt. Dann weisen Sie die Richtlinie der Gruppe `Consultants` zu. Wenn Sie die verwaltete Richtlinie einem Benutzer oder einer Gruppe zuordnen, erteilen Sie dem Benutzer oder der Gruppe die Berechtigung, eine Liste der Buckets abzurufen, die dem übergeordneten Benutzer gehören AWS-Konto.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.

Note

Da Sie Benutzerberechtigungen gewähren, melden Sie sich mit Ihrem AWS-Konto Anmeldeinformationen, nicht als IAM Benutzer.

2. Erstellen Sie die verwaltete Richtlinie.
 - a. Wählen Sie links im Navigationsbereich Policies (Richtlinien) und dann `Create Policy` (Richtlinie erstellen) aus.
 - b. Wählen Sie die JSONRegisterkarte.
 - c. Kopieren Sie die folgende Zugriffsrichtlinie und fügen Sie sie in das Textfeld für die Richtlinie ein.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowGroupToSeeBucketListInTheConsole",
    "Action": ["s3:ListAllMyBuckets"],
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::*"]
  }
]
```

Eine Richtlinie ist ein JSON Dokument. Im Dokument ist ein Statement ein Array von Objekten, die jeweils eine Berechtigung unter Verwendung einer Sammlung von Namenswertpaaren beschreiben. Die vorangegangene Richtlinie beschreibt eine bestimmte Berechtigung. Die Action definiert den Zugriffstyp. In der Richtlinie ist `s3:ListAllMyBuckets` eine vordefinierte Amazon-S3-Aktion. Diese Aktion deckt den Amazon S3 GET Service-Vorgang ab, der eine Liste aller Buckets zurückgibt, die dem authentifizierten Absender gehören. Der Wert des Effect-Elements bestimmt, ob die spezifische Berechtigung gewährt oder verweigert wird.

- d. Wählen Sie Review policy (Richtlinie überprüfen) aus. Geben Sie auf der nächsten Seite in das Feld Name `AllowGroupToSeeBucketListInTheConsole` ein und wählen Sie dann Create policy (Richtlinie erstellen).

Note

Der Eintrag Summary (Übersicht) enthält eine Nachricht, die angibt, dass die Richtlinie keinerlei Berechtigungen gewährt. Für diese Anleitung können Sie diese Nachricht getrost ignorieren.

3. Weisen Sie die von `AllowGroupToSeeBucketListInTheConsole` verwaltete Richtlinie, die Sie erstellt haben, der Gruppe `Consultants` zu.

step-by-step Anweisungen zum Anhängen einer verwalteten Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM Identitätsberechtigungen](#) im IAM Benutzerhandbuch.

In der IAM Konsole hängen Sie Richtliniendokumente an IAM Benutzer und Gruppen an. Da Sie möchten, dass beide Benutzer die Buckets auflisten können, weisen Sie die Richtlinie der Gruppe zu.

4. Die Berechtigung testen.

- a. Melden Sie sich über den IAM Benutzeranmeldelink (siehe [Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM](#)) mit beliebigen IAM Benutzeranmeldedaten an der Konsole an.
- b. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

Die Konsole sollte nun alle Buckets auflisten, nicht aber die Objekte in den jeweiligen Buckets.

Schritt 4.2: Benutzern gestatten, dass sie den Bucket-Inhalt auf Stammebene auflisten

Als Nächstes gestatten Sie allen Benutzern in der Gruppe `Consultants`, Elemente im Bucket `companybucket` aufzulisten. Wenn ein Benutzer den Unternehmensbucket in der Amazon-S3-Konsole auswählt, werden die Elemente in der Stammebene des Buckets angezeigt.

Note

Dieses Beispiel verwendet `companybucket` zur Veranschaulichung. Sie müssen den Namen des Bucket verwenden, den Sie erstellt haben.

Um die Anfrage zu verstehen, die die Konsole an Amazon S3 sendet, wenn Sie einen Bucket-Namen wählen, die Antwort, die Amazon S3 zurückgibt, und wie die Konsole die Antwort interpretiert, sollten Sie den Ablauf etwas genauer untersuchen.

Wenn Sie einen Bucket-Namen wählen, sendet die Konsole die [GETBucket-Anfrage \(List Objects\)](#) an Amazon S3. Diese Anforderung enthält die folgenden Parameter:

- Der Parameter `prefix` mit einer leeren Zeichenfolge als Wert.
- Der Parameter `delimiter` mit `/` als Wert.

Es folgt ein Beispiel einer Anforderung.

```
GET ?prefix=&delimiter=/ HTTP/1.1
Host: companybucket.s3.amazonaws.com
Date: Wed, 01 Aug 2012 12:00:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMblRepdf3YB+FIEXAMPLE=
```

Amazon S3 gibt eine Antwort zurück, die das folgende `<ListBucketResult/>`-Element enthält:


```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>companybucket</Name>
  <Prefix></Prefix>
  <Delimiter>/</Delimiter>
  ...
  <Contents>
    <Key>s3-dg.pdf</Key>
    ...
  </Contents>
  <CommonPrefixes>
    <Prefix>Development/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>Finance/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>Private/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Das Schlüsselobjekt `s3-dg.pdf` enthält nicht das Schrägstrich-Trennzeichen (`/`), und Amazon S3 gibt den Schlüssel im `<Contents>`-Element zurück. Alle anderen Schlüssel in unserem Beispiel-Bucket enthalten jedoch das `/`-Trennzeichen. Amazon S3 gruppiert diese Schlüssel und gibt ein einziges `<CommonPrefixes>`-Element für jeden der verschiedenen Präfix-Werte `Development/`, `Finance/` und `Private/` zurück, eine Unterzeichenfolge vom Anfang dieser Schlüssel bis zum ersten Auftreten des angegebenen `/`-Trennzeichens.

Die Konsole interpretiert dieses Ergebnis und zeigt die Elemente auf der Stammebene als drei Ordner und einen Objektschlüssel an.

Wenn Bob oder Alice den Ordner `Development` öffnen, sendet die Konsole die [GETBucket-Anfrage \(List Objects\)](#) an Amazon S3, wobei die `prefix delimiter` Parameter auf die folgenden Werte gesetzt sind:

- Der Parameter `prefix` mit dem Wert `Development/`.
- Der Parameter `delimiter` mit dem Wert `/`.

In der Antwort gibt Amazon S3 die Objektschlüssel mit dem angegebenen Präfix zurück.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
```

```

<Name>companybucket</Name>
<Prefix>Development</Prefix>
<Delimiter>/</Delimiter>
...
<Contents>
  <Key>Project1.xls</Key>
  ...
</Contents>
<Contents>
  <Key>Project2.xls</Key>
  ...
</Contents>
</ListBucketResult>

```

Die Konsole zeigt die Objektschlüssel an.

Kehren Sie nun wieder zu dem Vorgang zurück, Benutzern die Berechtigung zum Auflisten von Elementen auf der Stammebene zu erteilen. Damit der Bucket-Inhalt aufgelistet werden kann, müssen die Benutzer die Berechtigung zum Aufruf der `s3:ListBucket`-Aktion besitzen, wie in der folgenden Richtlinienanweisung dargestellt. Um sicherzustellen, dass sie nur den Inhalt auf der Stammebene sehen, können Sie als Bedingung hinzufügen, dass Benutzer in der Anforderung ein leeres `prefix` angeben müssen – d. h., dass es ihnen nicht gestattet sein soll, auf einen der Stammebenen-Ordner doppelzuklicken. Schließlich fügen Sie noch eine Bedingung hinzu, die den Zugriff im Ordnerstil dadurch vorschreibt, dass Benutzeranforderungen den Parameter `delimiter` mit dem Wert `/` enthalten müssen.

```

{
  "Sid": "AllowRootLevelListingOfCompanyBucket",
  "Action": ["s3:ListBucket"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition": {
    "StringEquals": {
      "s3:prefix":[""], "s3:delimiter":["/"]
    }
  }
}

```

Wenn Sie auf der Amazon S3 S3-Konsole einen Bucket auswählen, sendet die Konsole zunächst die [GETBucket-Standortanfrage](#), um den AWS-Region wo der Bucket bereitgestellt wird. Dann verwendet die Konsole den regionsspezifischen Endpunkt für den Bucket, um die Bucket-Anfrage

([GETList Objects](#)) zu senden. Wenn Benutzer die Konsole verwenden sollen, müssen Sie ihnen folglich die Berechtigung für die Aktion `s3:GetBucketLocation` gewähren, wie in der folgenden Richtlinienanweisung veranschaulicht.

```
{
  "Sid": "RequiredByS3Console",
  "Action": ["s3:GetBucketLocation"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3::*"]
}
```

Den Benutzern gestatten, den Bucket-Inhalt auf Stammebene aufzulisten

1. Melden Sie sich bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.

Benutze deine AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.

2. Ersetzen Sie die vorhandene `AllowGroupToSeeBucketListInTheConsole`-verwaltete Richtlinie, die der Gruppe `s3:ListBucket` zugeordnet ist, durch die folgende Richtlinie, durch die die Aktion `Consultants` ebenfalls gestattet wird. Denken Sie daran, zu ersetzen *Firma Bucket* in der Richtlinie `Resource` mit dem Namen Ihres Buckets.

step-by-step Anweisungen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinien bearbeiten](#). Wenn Sie den step-by-step Anweisungen folgen, achten Sie darauf, dass Sie die Schritte befolgen, um Ihre Änderungen auf alle wichtigen Entitäten anzuwenden, denen die Richtlinie zugeordnet ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
      "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
      "Action": [ "s3:ListAllMyBuckets", "s3:GetBucketLocation" ],
      "Effect": "Allow",
      "Resource": [ "arn:aws:s3::*" ]
    },
    {
      "Sid": "AllowRootLevelListingOfCompanyBucket",
```

```
"Action": ["s3:ListBucket"],
"Effect": "Allow",
"Resource": ["arn:aws:s3:::companybucket"],
"Condition":{
  "StringEquals":{
    "s3:prefix":[""], "s3:delimiter":["/"]
  }
}
]
```

3. Testen Sie die aktualisierten Berechtigungen.

- a. Melden Sie sich mithilfe des IAM Benutzeranmeldelinks (siehe [Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM](#)) bei AWS Management Console.

Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

- b. Wählen Sie den von Ihnen erstellten Bucket aus. Die Konsole zeigt die Bucket-Elemente auf Stammebene an. Wenn Sie Ordner im Bucket auswählen, können Sie den Ordnerinhalt nicht sehen, da Sie diese Berechtigungen noch nicht gewährt haben.

Dieser Test ist erfolgreich, wenn Benutzer die Amazon-S3-Konsole verwenden. Wenn Sie einen Bucket in der Konsole auswählen, sendet die Konsolenimplementierung eine Anforderung mit dem Parameter `prefix` mit einer leeren Zeichenfolge als Wert und dem Parameter `delimiter` mit `"/` als Wert.

Schritt 4.3: Übersicht über die Gruppenrichtlinie

Der Nettoeffekt der von Ihnen hinzugefügten Gruppenrichtlinie besteht darin, den IAM Benutzern Alice und Bob die folgenden Mindestberechtigungen zu gewähren:

- Auflisten aller Buckets, die dem übergeordneten Konto gehören.
- Ansicht der Elemente auf Stammebene im Bucket `companybucket`.

Die Benutzer können jedoch noch nicht sehr viele Aktionen ausführen. Als Nächstes erteilen Sie die folgenden benutzerspezifischen Berechtigungen:

- Erlauben Sie Bob, Objekte im Ordner `Development` aufzurufen und abzulegen.
- Erlauben Sie Bob, Objekte im Ordner `Finance` aufzurufen und abzulegen.

Für benutzerspezifische Berechtigungen fügen Sie eine Richtlinie zum spezifischen Benutzer hinzu, nicht für die Gruppe. Im folgenden Abschnitt erteilen Sie Alice die Berechtigung, mit dem Ordner `Development` zu arbeiten. Sie können die Schritte wiederholen, um Bob eine ähnliche Berechtigung für das Arbeiten im Ordner `Finance` zu erteilen.

Schritt 5: Erteilen Sie IAM der Benutzerin Alice spezifische Berechtigungen

Nun gewähren Sie Alice zusätzliche Berechtigungen, damit sie den Inhalt des Ordners `Development` aufrufen und Objekte in diesem Ordner ablegen kann.

Schritt 5.1: Erteilen Sie der IAM Benutzerin Alice die Erlaubnis, den Inhalt des Entwicklungsordners aufzulisten

Damit Alice den `Development` Ordnerinhalt auflisten kann, müssen Sie dem Benutzer Alice eine Richtlinie zuweisen, die die Berechtigung für die `s3:ListBucket` Aktion im `companybucket` Bucket erteilt, vorausgesetzt, die Anfrage enthält das Präfix `Development/`. Da diese Richtlinie nur auf die Benutzerin Alice angewendet werden soll, verwenden Sie eine Inline-Richtlinie. Weitere Informationen zu Inline-Richtlinien finden Sie unter [Verwaltete Richtlinien und Inline-Richtlinien](#) im IAMBenutzerhandbuch.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.

Benutze deine AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.

2. Erstellen Sie eine Inline-Richtlinie, um der Benutzerin Alice die Berechtigung zu erteilen, den Inhalt des Ordners `Development` aufzulisten.
 - a. Wählen Sie im Navigationsbereich auf der linken Seite `Users` (Benutzer) aus.
 - b. Wählen Sie den Benutzernamen `Alice`.
 - c. Wählen Sie auf der Benutzerdetailseite den Tab `Permissions` (Berechtigungen) und dann `Add inline policy` (Inline-Richtlinie hinzufügen).
 - d. Wählen Sie den `JSON` Tab.
 - e. Kopieren Sie die folgende Richtlinie und fügen Sie sie in das Richtlinientextfeld ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
    "Action": ["s3:ListBucket"],
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::companybucket"],
    "Condition":{
      "StringLike":{"s3:prefix":["Development/*"]}
    }
  }
]
}

```

- f. Wählen Sie Review policy (Richtlinie überprüfen) aus. Geben Sie auf der nächsten Seite in das Feld Name einen Namen ein und wählen Sie dann Create policy (Richtlinie erstellen).
3. Testen Sie die geänderten Berechtigungen für Alice:
 - a. Melden Sie sich mithilfe des IAM Benutzeranmeldelinks (siehe [Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM](#)) bei AWS Management Console.
 - b. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
 - c. Prüfen Sie in der Amazon-S3-Konsole, ob Alice die Liste der Objekte im Ordner Development/ im Bucket sehen kann.

Wenn die Benutzerin den Ordner /Development auswählt, um die Liste der darin enthaltenen Objekte anzuzeigen, sendet die Amazon-S3-Konsole die Anforderung ListObjects zusammen mit dem Präfix /Development an Amazon S3. Weil die Benutzerin die Erlaubnis besitzt, die Objektliste mit dem Präfix Development und dem Trennzeichen / zu sehen, gibt Amazon S3 die Objektliste mit dem Schlüsselpräfix Development/ zurück, und die Konsole gibt die Liste aus.

Schritt 5.2: Erteilen Sie der IAM Benutzerin Alice die Erlaubnis, Objekte abzurufen und im Entwicklungsordner abzulegen

Damit Alice Objekte im Ordner Development ablegen und aufrufen kann, benötigt sie die Berechtigung für die Aktionen s3:GetObject und s3:PutObject. Die folgenden Richtlinienanweisungen räumen diese Berechtigungen ein, vorausgesetzt die Anforderung enthält den Parameter prefix mit dem Wert Development/.

```

{
  "Sid": "AllowUserToReadWriteObjectData",
  "Action": ["s3:GetObject", "s3:PutObject"],
  "Effect": "Allow",

```

```
"Resource":["arn:aws:s3:::companybucket/Development/*"]
}
```

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

Verwenden Sie Ihre AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.

2. Bearbeiten Sie die Inline-Richtlinie, die Sie im vorherigen Schritt erstellt haben.
 - a. Wählen Sie im Navigationsbereich auf der linken Seite Users (Benutzer) aus.
 - b. Klicken Sie auf den Benutzernamen Alice.
 - c. Wählen Sie auf der Benutzerdetailseite den Tab Permissions (Berechtigungen) aus und erweitern Sie den Bereich Inline Policies (Inline-Richtlinien).
 - d. Wählen Sie neben dem Namen der im vorherigen Schritt erstellten Richtlinie Edit Policy (Richtlinie bearbeiten) aus.
 - e. Kopieren Sie die folgende Richtlinie und fügen Sie sie in das Textfeld für die Richtlinie ein, wobei die vorhandene Richtlinie ersetzt wird.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition": {
        "StringLike": {"s3:prefix": ["Development/*"]}
      }
    },
    {
      "Sid": "AllowUserToReadWriteObjectDataInDevelopmentFolder",
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket/Development/*"]
    }
  ]
}
```

```
}
```

3. Testen Sie die aktualisierte Richtlinie:
 - a. Melden Sie sich mithilfe des IAM Benutzeranmeldelinks (siehe [Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM](#)) bei AWS Management Console.
 - b. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
 - c. Prüfen Sie in der Amazon-S3-Konsole, ob Alice nun im Ordner Development ein Objekt hinzufügen oder herunterladen kann.

Schritt 5.3: Verweigern Sie der IAM Benutzerin Alice ausdrücklich die Berechtigungen für alle anderen Ordner im Bucket

Die Benutzerin kann jetzt den Inhalt auf Stammebene im Bucket companybucket auflisten. Sie kann auch Objekte im Ordner Development aufrufen und ablegen. Wenn Sie die Zugriffsberechtigungen weiter verbessern möchten, können Sie Alice den Zugriff auf andere Ordner im Bucket explizit verweigern. Wenn es eine andere Richtlinie (Bucket-Richtlinie oder ACL) gibt, die Alice Zugriff auf andere Ordner im Bucket gewährt, hat diese ausdrückliche Verweigerung Vorrang vor diesen Berechtigungen.

Sie können die folgende Anweisung zur Benutzerrichtlinie für Alice hinzufügen, die von allen Anfragen von Alice an Amazon S3 erfordert, dass der Parameter prefix enthalten ist, dessen Wert entweder Development/* oder eine leere Zeichenfolge ist.

```
{
  "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition":{
    "StringNotLike": {"s3:prefix":["Development/*",""] },
    "Null"           : {"s3:prefix":false }
  }
}
```

Beachten Sie, dass im Block Condition zwei bedingte Ausdrücke enthalten sind. Das Ergebnis dieser bedingten Ausdrücke wird durch das logische AND verknüpft. Wenn beide Bedingungen wahr sind, ist das Ergebnis der bedingten Ausdrücke wahr. Da der Wert für Effect in dieser Richtlinie Deny lautet, wenn Condition als "true" bewertet wird, können Benutzer die angegebene Action nicht durchführen.

- Der bedingte Ausdruck `Null` stellt sicher, dass die Anforderung von Alice den Parameter `prefix` enthält.

Der Parameter `prefix` erfordert einen ordnerartigen Zugriff. Wenn Sie eine Anfrage ohne den Parameter `prefix` senden, gibt Amazon S3 alle Objektschlüssel zurück.

Wenn die Anforderung den Parameter `prefix` mit einem Nullwert enthält, wird der Ausdruck als wahr ausgewertet, womit die gesamte Bedingung `Condition` wahr ist. Sie müssen eine leere Zeichenfolge für den Parameter `prefix` gestatten. Erinnern Sie sich an die vorangegangene Diskussion. Die leere Zeichenfolge lässt zu, dass Alice Bucket-Elemente auf Stammebene abrufen kann, wie es die Konsole in der vorhergehenden Diskussion macht. Weitere Informationen finden Sie unter [Schritt 4.2: Benutzern gestatten, dass sie den Bucket-Inhalt auf Stammebene auflisten](#).

- Der bedingte Ausdruck `StringNotLike` stellt sicher, dass die Anforderung scheitert, wenn der angegebene Wert des Parameters `prefix` nicht `Development/*` ist.

Folgen Sie den Schritten im vorherigen Abschnitt und aktualisieren Sie die Inline-Richtlinie noch einmal, die Sie für die Benutzerin Alice erstellt haben.

Kopieren Sie die folgende Richtlinie und fügen Sie sie in das Textfeld für die Richtlinie ein, wobei die vorhandene Richtlinie ersetzt wird.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition": {
        "StringLike": {"s3:prefix": ["Development/*"]}
      }
    },
    {
      "Sid": "AllowUserToReadWriteObjectDataInDevelopmentFolder",
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket/Development/*"]
    }
  ],
}
```

```
{
  "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition":{
    "StringNotLike": {"s3:prefix":["Development/*",""]} },
    "Null"           : {"s3:prefix":false }
  }
}
]
```

Schritt 6: Erteilen Sie dem IAM Benutzer Bob bestimmte Berechtigungen

Sie können die Schritte wiederholen, um Bob eine ähnliche Berechtigung für den Ordner Finance zu erteilen. Führen Sie die Schritte aus, mit denen Sie zuvor Alice Berechtigungen erteilt haben, aber ersetzen Sie den Ordner Development durch den Ordner Finance. step-by-step Anweisungen finden Sie unter [Schritt 5: Erteilen Sie IAM der Benutzerin Alice spezifische Berechtigungen](#).

Schritt 7: Absichern des Ordners „Private“

In diesem Beispiel haben Sie nur zwei Benutzer. Sie haben auf Gruppenebene alle erforderlichen Mindestberechtigungen erteilt und auf Benutzerebene die Berechtigungen nur dann gewährt, wenn die einzelnen Benutzer die Berechtigungen wirklich benötigen. Dieser Ansatz minimiert den Aufwand beim Verwalten der Berechtigungen. Wenn die Anzahl der Benutzer steigt, kann das Verwalten der Berechtigungen mühsam werden. Sie möchten z. B. nicht, dass irgendwelche der Benutzer in diesem Beispiel auf den Inhalt des Ordners Private zugreifen können. Wie stellen Sie sicher, dass Sie einem Benutzer nicht versehentlich Berechtigungen für den Private Ordner erteilen? Sie fügen eine Richtlinie hinzu, die explizit den Zugriff auf den Ordner verweigert. Eine explizite Zugriffsverweigerung überschreibt alle anderen Berechtigungen.

Um sicherzustellen, dass der Ordner Private auch privat bleibt, können Sie die folgenden beiden Ablehnungsanweisungen zur Gruppenrichtlinie hinzufügen:

- Fügen Sie die folgende Anweisung hinzu, um jede Aktion auf die Ressourcen im Private-Ordner (companybucket/Private/*) zu verweigern.

```
{
  "Sid": "ExplicitDenyAccessToPrivateFolderToEveryoneInTheGroup",
  "Action": ["s3:*"],
  "Effect": "Deny",
```

```
"Resource":["arn:aws:s3:::companybucket/Private/*"]
}
```

- Sie verweigern auch die Berechtigung für die Aktion Objekte auflisten, wenn die Anforderung das Präfix `Private/` angibt. Wenn Bob oder Alice in der Konsole den Order `Private` öffnen, bewirkt diese Richtlinie, dass Amazon S3 eine Fehlermeldung zurückgibt.

```
{
  "Sid": "DenyListBucketOnPrivateFolder",
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::*"],
  "Condition":{"
    "StringLike":{"s3:prefix":["Private/"]}
  }
}
```

Ersetzen Sie die Gruppenrichtlinie `Consultants` durch eine aktualisierte Richtlinie, die die vorherigen Ablehnungsanweisungen enthält. Nachdem die aktualisierte Richtlinie angewendet wurde, kann keiner der Benutzer in der Gruppe mehr auf den Order `Private` in Ihrem Bucket zugreifen.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

Verwenden Sie Ihre AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.

2. Ersetzen Sie die vorhandene von `AllowGroupToSeeBucketListInTheConsole` verwaltete Richtlinie, die der Gruppe `Consultants` zugeordnet ist, durch die folgende Richtlinie. Denken Sie daran, zu ersetzen *Firma Bucket* in der Richtlinie mit dem Namen Ihres Buckets.

Anweisungen finden Sie im IAMBenutzerhandbuch unter [Vom Kunden verwaltete Richtlinien bearbeiten](#). Wenn Sie die Anweisungen nachvollziehen, beachten Sie bitte die Anweisungen zum Ändern aller Hauptentitäten, denen die Richtlinie zugeordnet ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
      "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
```

```

    "Action": ["s3:ListAllMyBuckets", "s3:GetBucketLocation"],
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::*"]
  },
  {
    "Sid": "AllowRootLevelListingOfCompanyBucket",
    "Action": ["s3:ListBucket"],
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::companybucket"],
    "Condition":{
      "StringEquals":{"s3:prefix":[""]}
    }
  },
  {
    "Sid": "RequireFolderStyleList",
    "Action": ["s3:ListBucket"],
    "Effect": "Deny",
    "Resource": ["arn:aws:s3:::*"],
    "Condition":{
      "StringNotEquals":{"s3:delimiter":["/"]}
    }
  },
  {
    "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
    "Action": ["s3:*"],
    "Effect": "Deny",
    "Resource":["arn:aws:s3:::companybucket/Private/*"]
  },
  {
    "Sid": "DenyListBucketOnPrivateFolder",
    "Action": ["s3:ListBucket"],
    "Effect": "Deny",
    "Resource": ["arn:aws:s3:::*"],
    "Condition":{
      "StringLike":{"s3:prefix":["Private/"]}
    }
  }
]
}

```

Schritt 8: Bereinigen

Öffnen Sie zum Aufräumen die [IAMKonsole](#) und entfernen Sie die Benutzer Alice und Bob. step-by-step Anweisungen finden Sie im [IAMBenutzerhandbuch unter Löschen eines IAM Benutzers](#).

Um sicherzustellen, dass Sie für die Speicherung nicht künftig belastet werden, sollten Sie auch die Objekte und den Bucket löschen, die Sie für diese Übung erstellt haben.

Zugehörige Ressourcen

- [Verwaltung von IAM Richtlinien](#) im IAMBenutzerhandbuch

Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten

In diesem Thema werden die folgenden einführenden Anleitungsbeispiele für das Gewähren des Zugriffs auf Amazon-S3-Ressourcen bereitgestellt. Diese Beispiele verwenden die AWS Management Console um Ressourcen (Buckets, Objekte, Benutzer) zu erstellen und ihnen Berechtigungen zu erteilen. Anschließend zeigen die Beispiele auf, wie Sie Berechtigungen mithilfe der Befehlszeilen-Tools überprüfen können, sodass Sie keinen Code schreiben müssen. Wir stellen Befehle bereit, indem wir sowohl AWS Command Line Interface (AWS CLI) und die AWS Tools for Windows PowerShell.

- [Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen](#)

Die IAM Benutzer, die Sie in Ihrem Konto erstellen, haben standardmäßig keine Berechtigungen. In dieser Übung erteilen Sie einem Benutzer die Berechtigung, Bucket- und Objekt-Vorgänge auszuführen.

- [Beispiel 2: Bucket-Eigentümer erteilt kontoübergreifende Bucket-Berechtigungen](#)

In dieser Übung erteilt ein Bucket-Besitzer, Konto A, einem anderen Benutzer kontoübergreifende Berechtigungen AWS-Konto, Konto B. Konto B delegiert diese Berechtigungen dann an Benutzer in seinem Konto.

- Verwalten von Objektberechtigungen, wenn der Objekt- und der Bucket-Eigentümer nicht identisch sind

Bei den Beispielszenarien in diesem Fall geht es um einen Bucket-Eigentümer, der anderen Objektberechtigungen erteilt, es gehören jedoch nicht alle Objekte im Bucket dem Bucket-Eigentümer. Welche Berechtigungen benötigt der Bucket-Eigentümer und wie kann er diese Berechtigungen delegieren?

Das Tool AWS-Konto das einen Bucket erstellt, wird Bucket-Besitzer genannt. Der Besitzer kann andere gewähren AWS-Konten die Erlaubnis, Objekte hochzuladen, und AWS-Konten die Objekte erschaffen, besitzen sie. Der Bucket-Besitzer hat keine Rechte für Objekte, die von anderen erstellt wurden AWS-Konten. Wenn der Bucket-Besitzer eine Bucket-Richtlinie schreibt, die Zugriff auf Objekte gewährt, gilt die Richtlinie nicht für Objekte, die anderen Konten gehören.

In diesem Fall muss der Objekteigentümer dem Bucket-Besitzer zunächst mithilfe eines Objekts Berechtigungen erteilen ACL. Der Bucket-Besitzer kann diese Objektberechtigungen dann an

andere, an Benutzer in seinem eigenen Konto oder an einen anderen delegieren AWS-Konto, wie die folgenden Beispiele verdeutlichen.

- [Beispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören](#)

In dieser Übung erhält der Bucket-Eigentümer zuerst Berechtigungen von dem Objekteigentümer. Der Bucket-Eigentümer delegiert diese Berechtigungen anschließend an Benutzer in seinem eigenen Konto.

- [Beispiel 4: Bucket-Besitzer gewährt kontoübergreifende Berechtigungen für Objekte, die er nicht besitzt](#)

Nachdem der Bucket-Besitzer die Berechtigungen vom Objekteigentümer erhalten hat, kann er die Erlaubnis nicht an andere delegieren AWS-Konten weil die kontoübergreifende Delegierung nicht unterstützt wird (siehe [Berechtigungsdelegation](#)). Stattdessen kann der Bucket-Besitzer eine IAM Rolle mit Berechtigungen zum Ausführen bestimmter Operationen (z. B. zum Abrufen eines Objekts) erstellen und andere zulassen AWS-Konto um diese Rolle zu übernehmen. Jeder, der diese Rolle annimmt, kann anschließend auf Objekte zugreifen. Dieses Beispiel zeigt, wie ein Bucket-Besitzer eine IAM Rolle verwenden kann, um diese kontoübergreifende Delegierung zu aktivieren.

Bevor Sie die beispielhaften Walkthroughs ausprobieren

In diesen Beispielen wird der AWS Management Console um Ressourcen zu erstellen und Berechtigungen zu erteilen. Um Berechtigungen zu testen, verwenden die Beispiele die Befehlszeilentools, AWS CLI, und AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen. Zum Testen der Berechtigungen müssen Sie eins dieser Tools einrichten. Weitere Informationen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

Darüber hinaus verwenden diese Beispiele beim Erstellen von Ressourcen keine Root-Benutzeranmeldedaten eines AWS-Konto. Stattdessen erstellen Sie in diesen Konten einen Administratorbenutzer, um diese Aufgaben auszuführen.

Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen

AWS Identity and Access Management (IAM) empfiehlt, die Root-Benutzeranmeldedaten Ihres AWS-Konto um Anfragen zu stellen. Erstellen Sie stattdessen einen IAM Benutzer oder eine Rolle, gewähren Sie ihnen vollen Zugriff und verwenden Sie dann ihre Anmeldeinformationen, um Anfragen zu stellen. Wir bezeichnen dies als Administratorbenutzer oder -rolle. Weitere Informationen finden

Sie unter [Root-Benutzer des AWS-Kontos Anmeldeinformationen und IAM Identitäten](#) in der Allgemeine AWS-Referenz und [IAM bewährte Methoden](#) im IAM Benutzerhandbuch.

Alle Beispielanleitungen in diesem Abschnitt verwenden die Anmeldeinformationen des Administratorbenutzers. Wenn Sie noch keinen Administratorbenutzer für Ihr AWS-Konto, die Themen zeigen Ihnen, wie.

Um sich bei der anzumelden AWS Management Console Mit den Benutzeranmeldedaten müssen Sie sich mit der IAM Benutzeranmeldung anmelden URL. Die [IAM Konsole](#) bietet dies URL für Ihre AWS-Konto. Die Themen zeigen Ihnen, wie Sie die bekommen URL.

Einrichtung der Tools für die exemplarischen Vorgehensweisen

In den einführenden Beispielen (siehe [Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten](#)) wird der AWS Management Console um Ressourcen zu erstellen und Berechtigungen zu erteilen. Um Berechtigungen zu testen, verwenden die Beispiele die Befehlszeilentools, AWS Command Line Interface (AWS CLI) und AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen. Zum Testen der Berechtigungen müssen Sie eins dieser Tools einrichten.

Um das einzurichten AWS CLI

1. Laden Sie das herunter und konfigurieren Sie es AWS CLI. Anweisungen finden Sie in den folgenden Themen im AWS Command Line Interface Benutzerleitfaden:

[Installieren oder aktualisieren Sie auf die neueste Version von AWS Command Line Interface](#)

[Beginnen Sie mit dem AWS Command Line Interface](#)

2. Richten Sie das Standardprofil ein.

Sie speichern Benutzeranmeldedaten im AWS CLI Konfigurationsdatei. Erstellen Sie ein Standardprofil in der Konfigurationsdatei mit Ihrem AWS-Konto Anmeldeinformationen. Für Anweisungen zum Suchen und Bearbeiten Ihres AWS CLI Die Konfigurationsdatei finden Sie unter [Einstellungen für die Konfiguration und die Anmeldeinformationsdatei](#).

```
[default]
aws_access_key_id = access key ID
aws_secret_access_key = secret access key
region = us-west-2
```


3. Überprüfen Sie die Einrichtung, indem Sie den folgenden Befehl in die Befehlszeile eingeben. Beide Befehle stellen nicht explizit Anmeldeinformationen bereit, daher werden die Anmeldeinformationen des Standardprofils verwendet.

- Probieren Sie den `help` Befehl aus.

```
aws help
```

- Verwenden Sie den `aws s3 ls` Befehl, um eine Liste der Buckets für das konfigurierte Konto abzurufen.

```
aws s3 ls
```

Während Sie die exemplarischen Vorgehensweisen durchgehen, erstellen Sie Benutzer und speichern Benutzeranmeldeinformationen in den Konfigurationsdateien, indem Sie Profile erstellen, wie das folgende Beispiel zeigt. Diese Profile haben die Namen `AccountAdmin` und `AccountBadmin`.

```
[profile AccountAdmin]
aws_access_key_id = User AccountAdmin access key ID
aws_secret_access_key = User AccountAdmin secret access key
region = us-west-2

[profile AccountBadmin]
aws_access_key_id = Account B access key ID
aws_secret_access_key = Account B secret access key
region = us-east-1
```

Um mit diesen Benutzeranmeldeinformationen einen Befehl auszuführen, fügen Sie den Parameter `--profile` hinzu, um den Profilnamen festzulegen. Folgendes AWS CLI Befehl ruft eine Liste von Objekten ab *Beispiel/Bucket* und spezifiziert das `AccountBadmin` Profil.

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```


Alternativ können Sie eine Reihe von Anmeldeinformationen als Standardprofil konfigurieren, indem Sie die Umgebungsvariable `AWS_DEFAULT_PROFILE` von der Befehlszeile aus ändern. Nachdem du das getan hast, wann immer du auftrittst AWS CLI Befehle ohne den `--profile` Parameter,

die AWS CLI verwendet das Profil, das Sie in der Umgebungsvariablen festgelegt haben, als Standardprofil.

```
$ export AWS_DEFAULT_PROFILE=AccountAdmin
```

So führen Sie die Einrichtung durch: AWS Tools for Windows PowerShell

1. Laden Sie das herunter und konfigurieren Sie es AWS Tools for Windows PowerShell. Anweisungen finden Sie unter [Installation des AWS Tools for Windows PowerShell](#) in der AWS Tools for Windows PowerShell Benutzerleitfaden.

 Note

Um das zu laden AWS Tools for Windows PowerShell Modul müssen Sie aktivieren PowerShell Ausführung des Skripts. Weitere Informationen finden Sie unter [Aktivieren der Skriptausführung](#) im AWS Tools for Windows PowerShell Benutzerleitfaden.

2. Für diese exemplarischen Vorgehensweisen geben Sie Folgendes an AWS Anmeldeinformationen pro Sitzung mithilfe des `Set-AWSCredentials` Befehls. Der Befehl speichert die Anmeldeinformationen in einem persistenten Speicher (Parameter `-StoreAs`).

```
Set-AWSCredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas string
```

3. Überprüfen Sie die Einrichtung.

- Um eine Liste verfügbarer Befehle abzurufen, die Sie für Amazon S3 S3-Operationen verwenden können, führen Sie den `Get-Command` Befehl aus.

```
Get-Command -module awspowershell -noun s3* -StoredCredentials string
```

- Um eine Liste der Objekte in einem Bucket abzurufen, führen Sie den `Get-S3Object` Befehl aus.

```
Get-S3Object -BucketName bucketname -StoredCredentials string
```

Eine Liste der Befehle finden Sie unter [AWS Referenz zu Tools für PowerShell Cmdlets](#).

Jetzt sind Sie bereit, die exemplarischen Vorgehensweisen auszuprobieren. Folgen Sie den Links am Anfang jedes Abschnitts.

Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen

⚠ Important

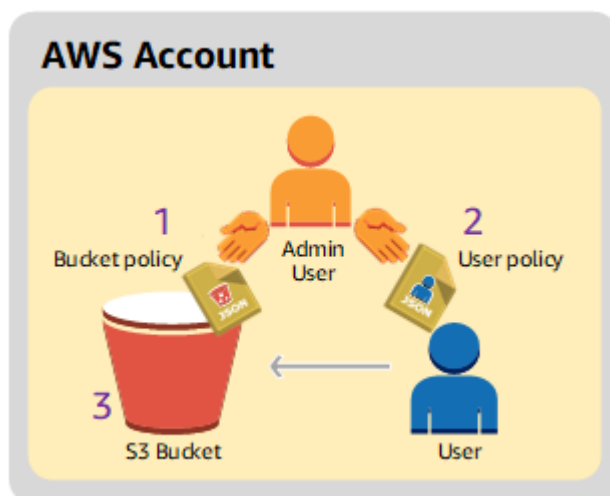
Das Erteilen von Berechtigungen für IAM Rollen ist eine bessere Methode als das Erteilen von Berechtigungen für einzelne Benutzer. Weitere Informationen darüber, wie Sie IAM Rollen Berechtigungen gewähren, finden Sie unter. [Grundlegendes zu kontoubergreifenden Berechtigungen und zur Verwendung von Rollen IAM](#)

Themen

- [Vorbereitung auf den Walkthrough](#)
- [Schritt 1: Erstellen Sie Ressourcen in Konto A und gewähren Sie Berechtigungen](#)
- [Schritt 2: Testen der Berechtigungen](#)

In dieser exemplarischen Vorgehensweise AWS-Konto besitzt einen Bucket, und das Konto umfasst einen IAM Benutzer. Standardmäßig hat der Benutzer keine Berechtigungen. Damit der Benutzer alle Aufgaben ausführen kann, muss das übergeordnete Konto ihm Berechtigungen erteilen. Der Bucket-Eigentümer und das übergeordnete Konto sind identisch. Um dem Benutzer also Berechtigungen für den Bucket zu gewähren, muss AWS-Konto kann eine Bucket-Richtlinie, eine Benutzerrichtlinie oder beides verwenden. Der Kontobesitzer gewährt einige Berechtigungen unter Verwendung einer Bucket-Richtlinie und andere Berechtigungen unter Verwendung einer Benutzerrichtlinie.

Die folgenden Schritte fasen das detaillierte Beispiel zusammen:



1. Der Kontoadministrator erstellt eine Bucket-Richtlinie, die dem Benutzer verschiedene Berechtigungen erteilt.
2. Der Kontoadministrator weist dem Benutzer eine Benutzerrichtlinie zu, die ihm zusätzliche Berechtigungen erteilt.
3. Anschließend probiert der Benutzer Berechtigungen aus, die über die Bucket-Richtlinie und die Benutzerrichtlinie erteilt wurden.

Für dieses Beispiel benötigen Sie eine AWS-Konto. Anstatt die Root-Benutzeranmeldedaten des Kontos zu verwenden, erstellen Sie einen Administratorbenutzer (siehe [Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen](#)). Wir beziehen uns auf AWS-Konto und den Administratorbenutzer, wie in der folgenden Tabelle dargestellt.

Konto-ID	Konto bezeichnet als	Administratorbenutzer im Konto
<i>1111-1111-1111</i>	Konto A	AccountAdmin

Note

Der Administratorbenutzer in diesem Beispiel ist AccountAdmin, was sich auf Konto A bezieht, und nicht AccountAdmin.

Alle Aufgaben zum Erstellen von Benutzern und zum Erteilen von Berechtigungen werden im AWS Management Console. Um die Berechtigungen zu überprüfen, werden in der exemplarischen Vorgehensweise die Befehlszeilentools verwendet. AWS Command Line Interface (AWS CLI) und AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen.

Vorbereitung auf den Walkthrough

1. Stellen Sie sicher, dass Sie eine haben AWS-Konto und dass es einen Benutzer mit Administratorrechten hat.
 - a. Melde dich an für ein AWS-Konto, falls erforderlich. Wir bezeichnen dieses Konto als Konto A.

- i. Gehe zu <https://aws.amazon.com/s3> und wähle **Create an AWS Konto**.
- ii. Folgen Sie den Anweisungen auf dem Bildschirm.

AWS wird Sie per E-Mail benachrichtigen, wenn Ihr Konto aktiv und für Sie verfügbar ist.

- b. Erstellen Sie in Konto A einen Administratorbenutzer **AccountAdmin**. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der [IAMKonsole](#) an und gehen Sie wie folgt vor:

- i. Erstellen Sie einen Benutzer **AccountAdmin** und notieren Sie sich die Sicherheitsanmeldedaten des Benutzers.

Anweisungen finden Sie unter [Einen IAM Benutzer erstellen in Ihrem AWS-Konto](#) im IAM-Benutzerhandbuch.

- ii. Gewähren Sie Administratorrechte, AccountAdmin indem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt.

Anweisungen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinien verwalten](#).

- iii. Notieren Sie sich die IAMBenutzeranmeldung URL für AccountAdmin. Sie müssen dies verwenden, URL wenn Sie sich bei der AWS Management Console. Weitere Informationen darüber, wo Sie die Anmeldung findenURL, finden Sie unter [Anmelden bei AWS Management Console als IAM Benutzer](#) im IAMBenutzerhandbuch. Beachten Sie die URL für jedes Konto.

2. Richten Sie entweder AWS CLI oder das AWS Tools for Windows PowerShell. Stellen Sie sicher, dass Sie die Administrator-Benutzeranmeldedaten wie folgt speichern:

- Wenn Sie das verwenden AWS CLI, erstellen Sie ein Profil,AccountAdmin, in der Konfigurationsdatei.
- Wenn Sie das verwenden AWS Tools for Windows PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als speichernAccountAdmin.

Detaillierte Anweisungen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

Schritt 1: Erstellen Sie Ressourcen in Konto A und gewähren Sie Berechtigungen

Melden Sie sich mit den Anmeldeinformationen des Benutzers AccountAdmin in Konto A und der speziellen IAM Benutzeranmeldung URL bei AWS Management Console und gehen Sie wie folgt vor:

1. Erstellen Sie Ressourcen für einen Bucket und einen IAM Benutzer
 - a. Erstellen Sie in der Amazon-S3-Konsole einen Bucket. Beachten Sie den AWS-Region in dem Sie den Bucket erstellt haben. Detaillierte Anweisungen finden Sie unter [Erstellen eines Buckets](#).
 - b. Gehen Sie in der [IAMKonsole](#) wie folgt vor:
 - i. Erstellen Sie einen Benutzer mit dem Namen Dave.

step-by-step Anweisungen finden Sie im Benutzerhandbuch unter [IAMBenutzer erstellen \(Konsole\)](#). IAM
 - ii. Notieren Sie sich die UserDave Anmeldeinformationen.
 - iii. Notieren Sie sich den Amazon-Ressourcennamen (ARN) für den Benutzer Dave. Wählen Sie in der [IAMKonsole](#) den Benutzer aus, und auf der Registerkarte Zusammenfassung wird der Benutzer angezeigtARN.

2. Erteilen Sie Berechtigungen.

Da der Bucket-Besitzer und das übergeordnete Konto, zu dem der Benutzer gehört, identisch sind AWS-Konto kann Benutzerberechtigungen mithilfe einer Bucket-Richtlinie, einer Benutzerrichtlinie oder beidem gewähren. In diesem Beispiel machen Sie beides. Wenn das Objekt auch demselben Konto gehört, kann der Bucket-Besitzer Objektberechtigungen in der Bucket-Richtlinie (oder einer IAM Richtlinie) gewähren.

- a. Fügen Sie in der Amazon S3 S3-Konsole die folgende Bucket-Richtlinie hinzu *awsexamplebucket1*.

Die Richtlinie enthält zwei Anweisungen.

- Die erste Anweisung erteilt Dave Berechtigungen für die Bucket-Operationen `s3:GetBucketLocation` und `s3:ListBucket`.
- Die zweite Anweisung erteilt die `s3:GetObject`-Berechtigung. Konto A gehört auch das Objekt, deshalb kann der Kontoadministrator die `s3:GetObject`-Berechtigung erteilen.

In der Principal Erklärung wird Dave von seinem Benutzer identifiziertARN. Weitere Informationen zu Richtlinienelementen finden Sie unter [Richtlinien und Berechtigungen in Amazon S3](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3::awsexamplebucket1"
      ]
    },
    {
      "Sid": "statement2",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3::awsexamplebucket1/*"
      ]
    }
  ]
}
```

- b. Erstellen Sie mithilfe der folgenden Richtlinie eine Inlinerichtlinie für den Benutzer Dave. Die Richtlinie erteilt dem Benutzer Dave die `s3:PutObject`-Berechtigung. Sie müssen die Richtlinie aktualisieren, indem Sie Ihren Bucket-Namen angeben.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PermissionForObjectOperations",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::awsexamplebucket1/*"
      ]
    }
  ]
}
```

Eine Anleitung dazu finden Sie IAMpolicies im IAMBenutzerhandbuch unter [Verwalten](#). Beachten Sie, dass Sie sich mit den Anmeldeinformationen von Konto A an der Konsole anmelden müssen.

Schritt 2: Testen der Berechtigungen

Überprüfen Sie unter Verwendung der Anmeldeinformationen von Dave, ob die Berechtigungen funktionieren. Sie haben die Wahl zwischen den folgenden beiden Verfahren.

Testen Sie die Berechtigungen mit dem AWS CLI

1. Aktualisieren Sie das AWS CLI Konfigurationsdatei, indem Sie das folgende UserDaveAccountA Profil hinzufügen. Weitere Informationen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Überprüfen Sie, ob Dave Vorgänge ausführen kann, für die ihm in der Benutzerrichtlinie Berechtigungen erteilt wurden. Laden Sie ein Beispielobjekt hoch, indem Sie Folgendes verwenden AWS CLI put-objectBefehl.

Der Parameter `--body` im Befehl identifiziert die hochzuladende Quelldatei. Befindet sich die Datei beispielsweise im Stammverzeichnis des Laufwerks C: auf einer Windows-Maschine, geben Sie den Pfad `:\HappyFace.jpg` an. Der Parameter `--key` gibt den Schlüsselnamen für das Objekt an.

```
aws s3api put-object --bucket awsexamplebucket1 --key HappyFace.jpg --  
body HappyFace.jpg --profile UserDaveAccountA
```

Führen Sie Folgendes als AWS CLI-Befehl aus, um das Objekt abzurufen.

```
aws s3api get-object --bucket awsexamplebucket1 --key HappyFace.jpg OutputFile.jpg  
--profile UserDaveAccountA
```

Testen Sie die Berechtigungen mit dem AWS Tools for Windows PowerShell

1. Speichern Sie Daves Anmeldeinformationen als `AccountADave`. Anschließend verwenden Sie diese Anmeldeinformationen, um PUT GET ein Objekt zu erstellen.

```
set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas  
AccountADave
```

2. Laden Sie ein Beispielobjekt hoch, indem Sie AWS Tools for Windows PowerShell `Write-S3Object`-Befehl verwenden, der die gespeicherten Anmeldeinformationen von Benutzer Dave verwendet.

```
Write-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file HappyFace.jpg  
-StoredCredentials AccountADave
```

Laden Sie das zuvor hochgeladene Objekt herunter.

```
Read-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file Output.jpg -  
StoredCredentials AccountADave
```

Beispiel 2: Bucket-Eigentümer erteilt kontoübergreifende Bucket-Berechtigungen

Important

Das Erteilen von Berechtigungen für IAM Rollen ist eine bessere Methode als das Erteilen von Berechtigungen für einzelne Benutzer. Weitere Informationen zur Vorgehensweise finden Sie unter [Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM](#).

Themen

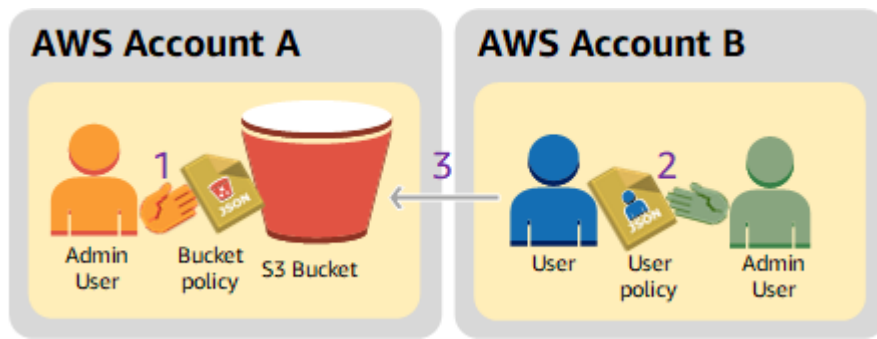
- [Vorbereitung auf den Walkthrough](#)
- [Schritt 1: Erledigen der Aufgaben von Konto A](#)
- [Schritt 2: Erledigen der Aufgaben von Konto B](#)
- [Schritt 3: \(Optional\) Versuchen Sie eine explizite Zugriffsverweigerung](#)
- [Schritt 4: Bereinigen](#)

Importieren in $\&S3$; AWS-Konto— zum Beispiel Konto A — kann einem anderen Konto gewährt werden AWS-Konto, Konto B, die Berechtigung, auf seine Ressourcen wie Buckets und Objekte zuzugreifen. Konto B kann diese Berechtigungen an Benutzer in seinem Konto delegieren. In diesem Beispielszenario erteilt ein Bucket-Eigentümer einem anderen Konto eine kontenübergreifende Berechtigung, um bestimmte Bucket-Vorgänge auszuführen.

Note

Konto A kann einem Benutzer in Konto B unter Verwendung einer Bucket-Richtlinie auch direkt Berechtigungen erteilen. Der Benutzer benötigt jedoch weiterhin die Erlaubnis des übergeordneten Kontos, Konto B, zu dem der Benutzer gehört, auch wenn Konto B keine Berechtigungen von Konto A hat. Solange der Benutzer sowohl vom Eigentümer der Ressource als auch vom übergeordneten Konto über die entsprechenden Berechtigungen verfügt, kann der Benutzer auf die Ressource zugreifen.

Nachfolgend finden Sie eine kurze Zusammenfassung der wichtigsten Details:



1. Der Administrator-Benutzer von Konto A ordnet eine Bucket-Richtlinie zu, die Konto B kontenübergreifende Berechtigungen erteilt, um bestimmte Bucket-Vorgänge auszuführen.

Beachten Sie, dass der Administrator-Benutzer in Konto B die Berechtigungen automatisch erbt.

2. Der Administrator-Benutzer von Konto B ordnet dem Benutzer eine Benutzerrichtlinie zu, die die Berechtigungen an den Benutzer delegiert, die er von Konto A erhalten hat.
3. Der Benutzer in Konto B überprüft die Berechtigungen, indem er auf ein Objekt in dem Bucket zugreift, das Konto A gehört.

Für dieses Beispiel benötigen Sie zwei Konten. Die folgende Tabelle zeigt, wie wir auf diese Konten und die Administrator-Benutzer darin verweisen. Gemäß den IAM Richtlinien (siehe [Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen](#)) verwenden wir in dieser exemplarischen Vorgehensweise keine Root-Benutzeranmeldedaten. Stattdessen erstellen Sie einen Administrator-Benutzer in jedem Konto, und verwenden dessen Anmeldeinformationen, um Ressourcen zu erstellen und ihnen Berechtigungen zu erteilen.

AWS-Konto ID	Konto bezeichnet als	Administratorbenutzer im Konto
<i>1111-1111-1111</i>	Konto A	AccountAdmin
<i>2222-2222-2222</i>	Konto B	AccountBAdmin

Alle Aufgaben zum Erstellen von Benutzern und zum Erteilen von Berechtigungen werden im AWS Management Console. Um die Berechtigungen zu überprüfen, werden in der exemplarischen Vorgehensweise die Befehlszeilentools verwendet. AWS Command Line Interface (CLI) und AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen.

Vorbereitung auf den Walkthrough

1. Stellen Sie sicher, dass Sie zwei haben AWS-Konten und dass jedes Konto einen Administratorbenutzer hat, wie in der Tabelle im vorherigen Abschnitt dargestellt.
 - a. Melden Sie sich an für ein AWS-Konto, falls erforderlich.
 - b. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der [IAMKonsole](#) an, um den Administratorbenutzer zu erstellen:
 - i. Erstellen Sie einen Benutzer **AccountAdmin** und notieren Sie sich die Sicherheitsanmeldedaten. Eine Anleitung finden Sie unter [Einen IAM Benutzer erstellen in Ihrem AWS-Konto](#) im IAM-Benutzerhandbuch.
 - ii. Gewähren Sie Administratorrechte, AccountAdmin indem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt. Anweisungen finden Sie im IAMBenutzerhandbuch unter [Arbeiten mit Richtlinien](#).
 - c. Notieren Sie sich die IAMBenutzeranmeldung URL auf dem Dashboard, während Sie sich in der IAM Konsole befinden. Alle Benutzer im Konto müssen dies verwenden, URL wenn sie sich bei der AWS Management Console.

Weitere Informationen finden Sie unter [So melden sich Benutzer bei Ihrem Konto an](#) im IAMBenutzerhandbuch.

- d. Wiederholen Sie den vorherigen Schritt mit den Anmeldeinformationen für Konto B und erstellen Sie einen Administratorbenutzer**AccountBadmin**.
2. Richten Sie entweder AWS Command Line Interface (AWS CLI) oder das AWS Tools for Windows PowerShell. Stellen Sie sicher, dass Sie die Administrator-Benutzeranmeldedaten wie folgt speichern:
 - Wenn Sie das verwenden AWS CLI, erstellen Sie zwei Profile AccountAdmin undAccountBadmin, in der Konfigurationsdatei.
 - Wenn Sie das verwenden AWS Tools for Windows PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als AccountAdmin und speichernAccountBadmin.

Detaillierte Anweisungen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

3. Speichern Sie die Anmeldeinformationen des Administrator-Benutzers, auch als Profile bezeichnet. Sie können den Profilnamen verwenden, statt für jeden eingegebenen Befehl

Anmeldeinformationen anzugeben. Weitere Informationen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

- a. Fügen Sie Profile hinzu in AWS CLI Anmeldeinformationsdatei für jeden der Administratorbenutzer AccountAdmin und AccountBadmin in den beiden Konten.

```
[AccountAdmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1

[AccountBadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1
```

- b. Wenn Sie das verwenden AWS Tools for Windows PowerShell, führen Sie den folgenden Befehl aus.

```
set-awscredentials -AccessKey AcctA-access-key-ID -SecretKey AcctA-secret-access-key -storeas AccountAdmin
set-awscredentials -AccessKey AcctB-access-key-ID -SecretKey AcctB-secret-access-key -storeas AccountBadmin
```

Schritt 1: Erledigen der Aufgaben von Konto A

Schritt 1.1: Melden Sie sich an AWS Management Console

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto A zunächst bei AWS Management Console als AccountAdminBenutzer. Dieser Benutzer erstellt einen Bucket und ordnet ihm eine Richtlinie zu.

Schritt 1.2: Erstellen eines Buckets

1. Erstellen Sie in der Amazon-S3-Konsole einen Bucket. In dieser Übung wird davon ausgegangen, dass der Bucket im Osten der USA (Nord-Virginia) erstellt wurde AWS-Region und ist benannt *amzn-s3-demo-bucket*.

Detaillierte Anweisungen finden Sie unter [Erstellen eines Buckets](#).

2. Hochladen eines Beispielobjekts in den Bucket.

Anweisungen finden Sie unter [Schritt 2: Hochladen eines Objekts in Ihren Bucket](#).

Schritt 1.3: Zuordnen einer Bucket-Richtlinie, um Konto B kontoübergreifende Berechtigungen zu erteilen

Die Bucket-Richtlinie gewährt Konto B die `s3:ListBucket` Berechtigungen `s3:GetLifecycleConfiguration` und. Es wird davon ausgegangen, dass Sie immer noch mit AccountAdminBenutzeranmeldedaten bei der Konsole angemeldet sind.

1. Weisen Sie *amzn-s3-demo-bucket* die folgende Bucket-Richtlinie zu. Die Richtlinie erteilt Konto B die Berechtigung für die Aktionen `s3:GetLifecycleConfiguration` und `s3:ListBucket`.

Detaillierte Anweisungen finden Sie unter [Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Example permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:root"
      },
      "Action": [
        "s3:GetLifecycleConfiguration",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3::amzn-s3-demo-bucket"
      ]
    }
  ]
}
```

2. Stellen Sie sicher, dass Konto B (und damit sein Administratorbenutzer) die Vorgänge ausführen kann.
 - Überprüfen Sie mit dem AWS CLI

```
aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmin
aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --
profile AccountBadmin
```

- Überprüfen Sie mit dem AWS Tools for Windows PowerShell

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBadmin
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -
StoredCredentials AccountBadmin
```

Schritt 2: Erledigen der Aufgaben von Konto B

Jetzt erstellt der Administrator von Konto B einen Benutzer Dave und delegiert an Dave die Berechtigungen, die er von Konto A erhalten hat.

Schritt 2.1: Melden Sie sich an bei AWS Management Console

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto B zunächst bei AWS Management Console als AccountBadminBenutzer.

Schritt 2.2: Erstellen des Benutzers Dave in Konto B

Erstellen Sie in der [IAMKonsole](#) einen Benutzer, **Dave**.

Anweisungen finden Sie im Benutzerhandbuch unter [IAMBenutzer erstellen \(Konsole\)](#). IAM

Schritt 2.3: Delegieren von Berechtigungen an den Benutzer Dave

Erstellen Sie mithilfe der folgenden Richtlinie eine Inlinerichtlinie für den Benutzer Dave. Sie müssen die Richtlinie aktualisieren, indem Sie Ihren Bucket-Namen angeben.

Es wird davon ausgegangen, dass Sie mit AccountBadminBenutzeranmeldedaten bei der Konsole angemeldet sind.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Example",
```



```
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket"
    ]
  }
]
```

Anweisungen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinien verwalten](#).

Schritt 2.4: Testen der Berechtigungen

Jetzt kann Dave in Konto B den Inhalt von *amzn-s3-demo-bucket* auflisten, der Konto A gehört. Sie können die Berechtigungen mit einem der folgenden Verfahren überprüfen.

Testen Sie die Berechtigungen mit dem AWS CLI

1. Fügen Sie das UserDave Profil dem hinzu AWS CLI Konfigurationsdatei. Weitere Informationen zur Config-Datei finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

```
[profile UserDave]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Geben Sie in der Befehlszeile Folgendes ein AWS CLI Befehl, um zu überprüfen, ob Dave jetzt eine Objektliste aus dem *amzn-s3-demo-bucket* Konto A abrufen kann. Beachten Sie, dass der Befehl das UserDave Profil angibt.

```
aws s3 ls s3://amzn-s3-demo-bucket --profile UserDave
```

Dave hat keine anderen Berechtigungen. Wenn er also eine andere Operation versucht, z. B. die folgende `get-bucket-lifecycle-configuration` Konfiguration, gibt Amazon S3 die Berechtigung verweigert zurück.

```
aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --profile UserDave
```

Testen Sie die Berechtigungen mit AWS Tools for Windows PowerShell

1. Speichern Sie Daves Anmeldeinformationen als `AccountBDave`.

```
set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas  
AccountBDave
```

2. Probieren Sie den Befehl `List Bucket` aus.

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
```

Dave hat keine anderen Berechtigungen. Wenn er also einen anderen Vorgang versucht, z. B. den folgenden, gibt `get-s3bucketlifecycleconfiguration` Amazon S3 die Berechtigung verweigert zurück.

```
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -  
StoredCredentials AccountBDave
```

Schritt 3: (Optional) Versuchen Sie eine explizite Zugriffsverweigerung

Sie können sich Berechtigungen mithilfe einer Zugriffskontrollliste (ACL), einer Bucket-Richtlinie oder einer Benutzerrichtlinie gewähren lassen. Wenn jedoch entweder durch eine Bucket-Richtlinie oder eine Benutzerrichtlinie eine ausdrückliche Verweigerung festgelegt wird, hat die ausdrückliche Verweigerung Vorrang vor allen anderen Berechtigungen. Aktualisieren Sie zum Testen die Bucket-Richtlinie und verweigern Sie Konto B ausdrücklich die `s3:ListBucket` Erlaubnis. Die Richtlinie gewährt auch die `s3:ListBucket` Erlaubnis. Die ausdrückliche Ablehnung hat jedoch Vorrang, und Konto B oder Benutzer in Konto B können keine Objekte auflisten. *amzn-s3-demo-bucket*

1. Wenn Sie die Anmeldeinformationen des Benutzers `AccountAdmin` in Konto A verwenden, ersetzen Sie die Bucket-Richtlinie durch Folgendes.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Example permissions",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::AccountB-ID:root"      }  
    }  
  ]  
}
```

```
    },
    "Action": [
      "s3:GetLifecycleConfiguration",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket"
    ]
  },
  {
    "Sid": "Deny permission",
    "Effect": "Deny",
    "Principal": {
      "AWS": "arn:aws:iam::AccountB-ID:root"
    },
    "Action": [
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket"
    ]
  }
]
```

2. Wenn Sie nun versuchen, mithilfe von AccountBadmIn Anmeldeinformationen eine Bucket-Liste zu erstellen, wird der Zugriff verweigert.

- Verwendung der AWS CLI, führen Sie den folgenden Befehl aus:

```
aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmIn
```

- Verwendung der AWS Tools for Windows PowerShell, führen Sie den folgenden Befehl aus:

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
```


Schritt 4: Bereinigen

1. Nachdem Sie mit dem Testen fertig sind, können Sie zum Aufräumen wie folgt vorgehen:

- Melden Sie sich an bei AWS Management Console ([AWS Management Console](#)) verwenden Sie die Anmeldeinformationen von Konto A und gehen Sie wie folgt vor:

- Entfernen Sie in der Amazon-S3-Konsole die an *amzn-s3-demo-bucket* angefügte Bucket-Richtlinie. Löschen Sie in den Bucket Properties die Richtlinie im Abschnitt Permissions.
 - Wenn der Bucket für diese Übung erstellt wurde, löschen Sie in der Amazon-S3-Konsole die Objekte und dann den Bucket.
 - Entfernen Sie den AccountAdmin Benutzer in der [IAMKonsole](#).
2. Melden Sie sich mit den Anmeldeinformationen für Konto B bei der [IAMKonsole](#) an. Benutzer löschenAccountAdmin. step-by-step Anweisungen finden Sie im [IAMBenutzerhandbuch unter Löschen eines IAM](#) Benutzers.

Beispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören

 **Important**

Das Erteilen von Berechtigungen für IAM Rollen ist eine bessere Methode als das Erteilen von Berechtigungen für einzelne Benutzer. Weitere Informationen zur Vorgehensweise finden Sie unter [Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM](#).

Themen

- [Schritt 0: Vorbereitung auf den Walkthrough](#)
- [Schritt 1: Erledigen der Aufgaben von Konto A](#)
- [Schritt 2: Erledigen der Aufgaben von Konto B](#)
- [Schritt 3: Testen der Berechtigungen](#)
- [Schritt 4: Bereinigen](#)

Das Szenario für dieses Beispiel ist, dass ein Bucket-Besitzer die Erlaubnis zum Zugriff auf Objekte gewähren möchte, aber der Bucket-Besitzer besitzt nicht alle Objekte im Bucket. In diesem Beispiel versucht der Bucket-Eigentümer, Benutzern in seinem eigenen Konto eine Berechtigung zu erteilen.

Ein Bucket-Besitzer kann andere aktivieren AWS-Konten um Objekte hochzuladen. Standardmäßig besitzt der Bucket-Besitzer keine Objekte, die von einem anderen in einen Bucket geschrieben wurden AWS-Konto. Objekte gehören den Konten, die sie in einen S3-Bucket schreiben. Wenn der

Bucket-Besitzer keine Objekte im Bucket besitzt, muss der Objekteigentümer dem Bucket-Besitzer zunächst mithilfe einer Objektzugriffskontrollliste (ACL) die Erlaubnis erteilen. Anschließend kann der Bucket-Besitzer einem Objekt, das er nicht besitzt, Berechtigungen gewähren. Weitere Informationen finden Sie unter [Amazon-S3-Bucket- und Objekt-Eigentümerschaft](#).

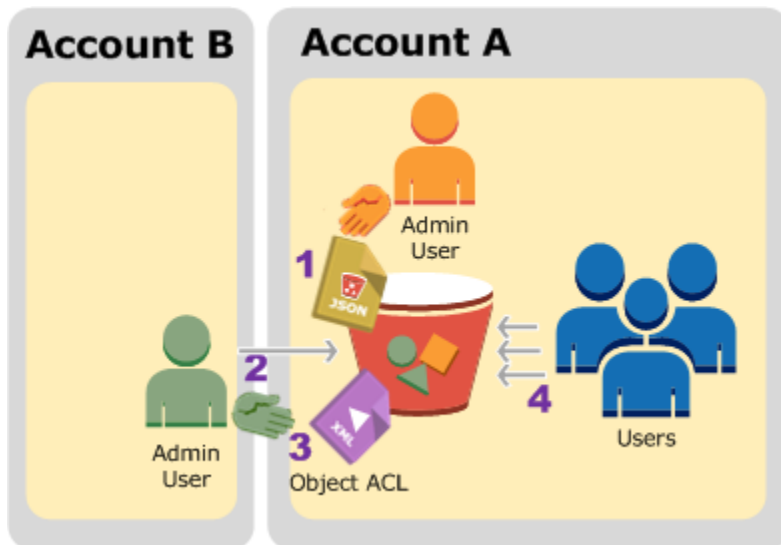
Wenn der Bucket-Besitzer die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum auf den Bucket anwendet, besitzt der Bucket-Besitzer alle Objekte im Bucket, einschließlich der Objekte, die von einem anderen geschrieben wurden AWS-Konto. Dieser Ansatz behebt das Problem, dass Objekte nicht dem Bucket-Besitzer gehören. Anschließend können Sie Berechtigungen an Benutzer in Ihrem eigenen Konto oder an andere Benutzer delegieren AWS-Konten.

Note

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

In diesem Beispiel gehen wir davon aus, dass der Bucket-Eigentümer die vom Bucket-Eigentümer erzwungene Einstellung für Object Ownership nicht angewendet hat. Der Bucket-Eigentümer delegiert die Berechtigung an Benutzer in seinem eigenen Konto. Im Folgenden finden Sie eine Zusammenfassung der exemplarischen Schritte:



1. Der Administrator-Benutzer von Konto A ordnet eine Bucket-Richtlinie mit zwei Anweisungen zu.
 - Kontoübergreifende Berechtigung für Konto B, um Objekte hochzuladen.
 - Berechtigung für einen Benutzer in seinem eigenen Konto, auf Objekte im Bucket zuzugreifen.
2. Der Administrator-Benutzer für Konto B lädt Objekte in den Bucket hoch, der Konto A gehört.
3. Der Administrator für Konto B aktualisiert die Zuweisung ACL zum Hinzufügen von Objekten, die dem Bucket-Besitzer die volle Kontrolle über das Objekt gewährt.
4. Der Benutzer in Konto A überprüft dies durch Zugriff auf Objekte im Bucket, unabhängig davon, wem diese gehören.

Für dieses Beispiel benötigen Sie zwei Konten. Die folgende Tabelle zeigt, wie wir auf diese Konten und die Administrator-Benutzer in diesen Konten verweisen. In dieser exemplarischen Vorgehensweise verwenden Sie die Root-Benutzeranmeldedaten des Kontos nicht gemäß den empfohlenen IAM Richtlinien. Weitere Informationen finden Sie unter [Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen](#). Stattdessen erstellen Sie einen Administrator in jedem Konto, und verwenden dessen Anmeldeinformationen, um Ressourcen zu erstellen und ihnen Berechtigungen zu erteilen.

AWS-Konto ID	Konto bezeichnet als	Administrator im Konto
<i>1111-1111-1111</i>	Konto A	AccountAdmin
<i>2222-2222-2222</i>	Konto B	AccountBadmin

Alle Aufgaben zum Erstellen von Benutzern und zum Erteilen von Berechtigungen werden im AWS Management Console. Um die Berechtigungen zu überprüfen, werden in der exemplarischen Vorgehensweise die Befehlszeilentools verwendet. AWS Command Line Interface (AWS CLI) und AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen.

Schritt 0: Vorbereitung auf den Walkthrough

1. Stellen Sie sicher, dass Sie zwei haben AWS-Konten und jedes Konto hat einen Administrator, wie in der Tabelle im vorherigen Abschnitt gezeigt.
 - a. Melden Sie sich an für ein AWS-Konto, falls erforderlich.
 - b. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der [IAMKonsole](#) an und gehen Sie wie folgt vor, um einen Administratorbenutzer zu erstellen:
 - Erstellen Sie einen Benutzer **AccountAdmin** und notieren Sie sich die Sicherheitsanmeldedaten des Benutzers. Weitere Informationen zum Hinzufügen von Benutzern finden Sie unter [Einen IAM Benutzer erstellen in Ihrem AWS-Konto](#) im IAM-Benutzerhandbuch.
 - Gewähren Sie Administratorberechtigungen für, AccountAdmin indem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt. Anweisungen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinien verwalten](#).
 - Notieren Sie sich im [IAMKonsolen-Dashboard](#) die IAMBenutzeranmeldung URL. Benutzer in diesem Konto müssen dies verwendenURL, wenn sie sich bei der AWS Management Console. Weitere Informationen finden Sie [im Benutzerhandbuch unter So melden sich IAM Benutzer bei Ihrem Konto](#) an.
 - c. Wiederholen Sie den vorherigen Schritt mit den Anmeldeinformationen für Konto B und erstellen Sie einen Administratorbenutzer**AccountBadmin**.
2. Richten Sie entweder AWS CLI oder die Tools für Windows PowerShell. Stellen Sie sicher, dass Sie die Administratoranmeldeinformationen wie folgt speichern:
 - Wenn Sie das verwenden AWS CLI, erstellen Sie zwei Profile AccountAdmin undAccountBadmin, in der Konfigurationsdatei.
 - Wenn Sie die Tools für Windows verwenden PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als AccountAdmin und speichernAccountBadmin.

Detaillierte Anweisungen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

Schritt 1: Erledigen der Aufgaben von Konto A

Führen Sie für Konto A die folgenden Schritte aus:

Schritt 1.1: Anmelden bei der Konsole

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto A bei AWS Management Console als **AccountAdmin** Benutzer. Dieser Benutzer erstellt einen Bucket und ordnet ihm eine Richtlinie zu.

Schritt 1.2: Erstellen eines Buckets und eines Benutzers und Hinzufügen einer Bucket-Richtlinie, um Benutzerberechtigungen zu erteilen

1. Erstellen Sie in der Amazon-S3-Konsole einen Bucket. In dieser Übung wird davon ausgegangen, dass der Bucket im Osten der USA (Nord-Virginia) erstellt wurde AWS-Region, und der Name ist *amzn-s3-demo-bucket1*.

Detaillierte Anweisungen finden Sie unter [Erstellen eines Buckets](#).

2. Erstellen Sie in der [IAMKonsole](#) einen Benutzer **Dave**.

step-by-step Anweisungen finden Sie im Benutzerhandbuch unter [IAMBenutzer erstellen \(Konsole\)](#). IAM

3. Notieren Sie sich die Anmeldeinformationen des Benutzers Dave.
4. Weisen Sie in der Amazon-S3-Konsole die folgende Bucket-Richtlinie dem *amzn-s3-demo-bucket1*-Bucket zu. Detaillierte Anweisungen finden Sie unter [Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole](#). Folgen Sie den Schritten, um eine Bucket-Richtlinie hinzuzufügen. Informationen darüber, wie Sie ein Konto findenIDs, finden Sie unter [Finden Ihres AWS-Konto Ausweis](#).

Die Richtlinie erteilt Konto B die Berechtigung `s3:PutObject` und `s3:ListBucket`. Die Richtlinie gewährt Dave dem Benutzer auch die `s3:GetObject` Erlaubnis.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:root"
      }
    }
  ],
```



```

    "Action": [
      "s3:PutObject",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket1/*",
      "arn:aws:s3:::amzn-s3-demo-bucket1"
    ]
  },
  {
    "Sid": "Statement3",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
    },
    "Action": [
      "s3:GetObject"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket1/*"
    ]
  }
]
}

```

Schritt 2: Erledigen der Aufgaben von Konto B

Da Konto B nun über die erforderlichen Berechtigungen verfügt, um Vorgänge im Bucket von Konto A durchzuführen, geht der Administrator von Konto B wie folgt vor:

- Lädt ein Objekt in den Bucket von Konto A hoch
- Fügt dem Objekt einen Grant hinzuACL, um Account A, dem Bucket-Besitzer, die volle Kontrolle zu gewähren

Verwendung der AWS CLI

1. Verwendung der `put-object` AWS CLI Befehl, lade ein Objekt hoch. Der Parameter `--body` im Befehl identifiziert die hochzuladende Quelldatei. Wenn sich die Datei beispielsweise auf dem C: Laufwerk eines befindet Windows Maschine, spezifizierenc: `\HappyFace.jpg`. Der Parameter `--key` gibt den Schlüsselnamen für das Objekt an.

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --body  
HappyFace.jpg --profile AccountBadmin
```

2. Fügen Sie dem Objekt einen Grant hinzuACL, um dem Bucket-Besitzer die volle Kontrolle über das Objekt zu gewähren. Informationen darüber, wie Sie eine kanonische Benutzer-ID finden Sie unter [Finden Sie die kanonische Benutzer-ID für Ihr AWS-Konto](#) in der AWS Referenzhandbuch zur Kontoverwaltung.

```
aws s3api put-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --grant-  
full-control id="AccountA-CanonicalUserID" --profile AccountBadmin
```

Verwenden der Tools für Windows PowerShell

1. Laden Sie mit dem Write-S3Object Befehl ein Objekt hoch.

```
Write-S3Object -BucketName amzn-s3-demo-bucket1 -key HappyFace.jpg -file  
HappyFace.jpg -StoredCredentials AccountBadmin
```

2. Fügen Sie dem Objekt einen Grant hinzuACL, um dem Bucket-Besitzer die volle Kontrolle über das Objekt zu gewähren.

```
Set-S3ACL -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -CannedACLName  
"bucket-owner-full-control" -StoredCreden
```

Schritt 3: Testen der Berechtigungen

Überprüfen Sie nun, dass der Benutzer Dave in Konto A Zugriff auf das Objekt hat, das Konto B gehört.

Verwendung der AWS CLI

1. Fügen Sie die Anmeldeinformationen des Benutzers Dave zur AWS CLI Konfigurationsdatei und erstellen Sie ein neues Profil, UserDaveAccountA. Weitere Informationen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

```
[profile UserDaveAccountA]  
aws_access_key_id = access-key  
aws_secret_access_key = secret-access-key
```

```
region = us-east-1
```

2. Führen Sie den `get-object` CLI Befehl aus, um ihn herunterzuladen `HappyFace.jpg` und lokal zu speichern. Sie stellen dem Benutzer Dave Anmeldeinformationen bereit, indem Sie den Parameter `--profile` hinzufügen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key  
HappyFace.jpg Outputfile.jpg --profile UserDaveAccountA
```

Verwenden der Tools für Windows PowerShell

1. Speichere den Benutzer Dave AWS Anmeldeinformationen, wie `UserDaveAccountA`, für den persistenten Speicher.

```
Set-AWSCredentials -AccessKey UserDave-AccessKey -SecretKey UserDave-  
SecretAccessKey -storeas UserDaveAccountA
```

2. Führen Sie den Befehl `Read-S3Object` aus, um das Objekt `HappyFace.jpg` herunterzuladen und es lokal zu speichern. Sie stellen dem Benutzer Dave Anmeldeinformationen bereit, indem Sie den Parameter `-StoredCredentials` hinzufügen.

```
Read-S3Object -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -file  
HappyFace.jpg -StoredCredentials UserDaveAccountA
```

Schritt 4: Bereinigen

1. Nachdem Sie mit dem Testen fertig sind, können Sie zum Aufräumen wie folgt vorgehen:
 - Melden Sie sich an bei [AWS Management Console](#) verwenden Sie die Anmeldeinformationen von Konto A und gehen Sie wie folgt vor:
 - Entfernen Sie in der Amazon S3 S3-Konsole die Bucket-Richtlinie, die angehängt ist an *amzn-s3-demo-bucket1*. Löschen Sie im Bucket Properties die Richtlinie im Abschnitt Permissions.
 - Wenn der Bucket für diese Übung erstellt wurde, löschen Sie in der Amazon-S3-Konsole die Objekte und dann den Bucket.
 - Entfernen Sie den AccountAdminBenutzer in der [IAMKonsole](#). step-by-step Anweisungen finden Sie im [IAMBenutzerhandbuch unter Löschen eines IAM Benutzers](#).

2. Melden Sie sich bei [AWS Management Console](#) mit den Anmeldeinformationen für Konto B. Löschen Sie den Benutzer in der [IAM Konsole Account Admin](#).

Beispiel 4: Bucket-Besitzer gewährt kontoübergreifende Berechtigungen für Objekte, die er nicht besitzt

Themen

- [Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM](#)
- [Schritt 0: Vorbereitung auf den Walkthrough](#)
- [Schritt 1: Erledigen der Aufgaben von Konto A](#)
- [Schritt 2: Erledigen der Aufgaben von Konto B](#)
- [Schritt 3: Führen Sie die Aufgaben von Konto C aus](#)
- [Schritt 4: Bereinigen](#)
- [Zugehörige Ressourcen](#)

In diesem Beispielszenario besitzen Sie einen Bucket und haben andere aktiviert AWS-Konten um Objekte hochzuladen. Wenn Sie die vom Bucket-Besitzer erzwungene Einstellung für S3 Object Ownership auf den Bucket angewendet haben, gehören Ihnen alle Objekte im Bucket, einschließlich der Objekte, die von einem anderen geschriebenen wurden AWS-Konto. Dieser Ansatz behebt das Problem, dass Objekte nicht Ihnen, dem Bucket-Besitzer, gehören. Anschließend können Sie Berechtigungen an Benutzer in Ihrem eigenen Konto oder an andere Benutzer delegieren AWS-Konten. Angenommen, die vom Bucket-Besitzer erzwungene Einstellung für S3 Object Ownership ist nicht aktiviert. Das heißt, Ihr Bucket kann Objekte enthalten, die andere AWS-Konten besitzen.

Angenommen, Sie müssen als Bucket-Eigentümer einem Benutzer in einem anderen Konto eine kontenübergreifende Berechtigung für Objekte erteilen, unabhängig davon, wer der Eigentümer ist. Dieser Benutzer könnte beispielsweise eine Buchhaltungsanwendung sein, die Zugriff auf Objekt-Metadaten benötigt. Es gibt zwei Kernprobleme:

- Der Bucket-Besitzer hat keine Rechte für Objekte, die von anderen erstellt wurden AWS-Konten. Damit der Bucket-Besitzer Berechtigungen für Objekte gewähren kann, die er nicht besitzt, muss der Objekteigentümer zuerst dem Bucket-Besitzer die entsprechenden Rechte erteilen. Der Objekteigentümer ist AWS-Konto der die Objekte erstellt hat. Der Bucket-Eigentümer kann diese Berechtigungen delegieren.

- Das Bucket-Eigentümerkonto kann Berechtigungen an Benutzer in seinem eigenen Konto delegieren (siehe [Beispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören](#)). Das Konto des Bucket-Besitzers kann jedoch keine Berechtigungen an andere delegieren AWS-Konten weil die kontoübergreifende Delegierung nicht unterstützt wird.

In diesem Szenario kann der Bucket-Besitzer eine erstellen AWS Identity and Access Management (IAM) Rolle mit der Berechtigung zum Zugriff auf Objekte Dann kann der Bucket-Besitzer eine weitere Rolle gewähren AWS-Konto die Erlaubnis, die Rolle zu übernehmen, wodurch sie vorübergehend auf Objekte im Bucket zugreifen kann.

Note

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM

IAM Rollen ermöglichen verschiedene Szenarien, um den Zugriff auf Ihre Ressourcen zu delegieren, und der kontenübergreifende Zugriff ist eines der wichtigsten Szenarien. In diesem Beispiel verwendet der Bucket-Besitzer, Konto A, eine IAM Rolle, um vorübergehend den kontoübergreifenden Objektzugriff an Benutzer in einem anderen zu delegieren AWS-Konto, Konto C. Jeder IAM Rolle, die Sie erstellen, sind die folgenden zwei Richtlinien zugeordnet:

- Eine Vertrauensrichtlinie, die eine andere identifiziert AWS-Konto das kann die Rolle übernehmen.

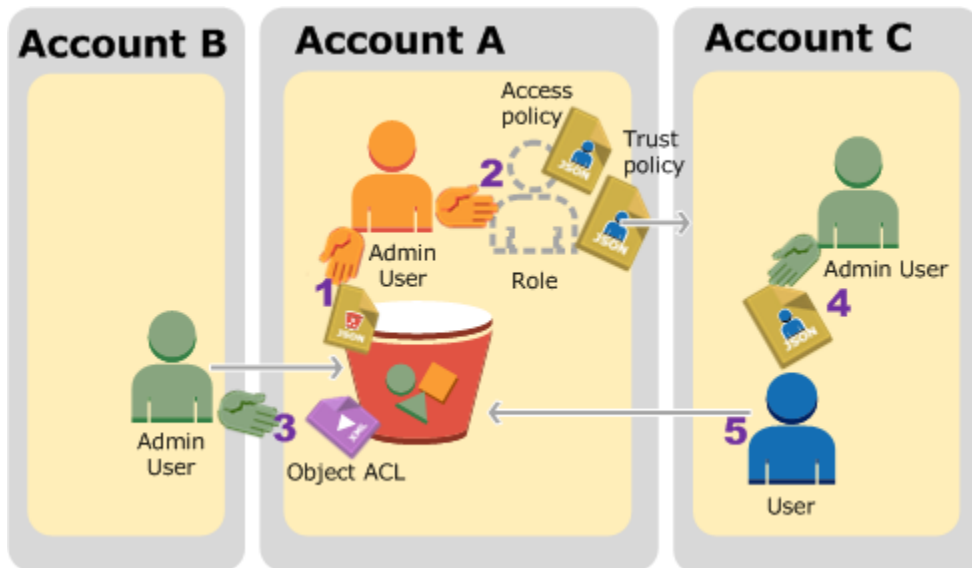
- Eine Zugriffsrichtlinie, die definiert, welche Berechtigungen – z. B. `s3:GetObject` – zulässig sind, wenn jemand die Rolle einnimmt. Eine Liste aller Berechtigungen, die Sie in einer Richtlinie angeben können, finden Sie unter [Politische Maßnahmen für Amazon S3](#).

Das Tool AWS-Konto Die in der Vertrauensrichtlinie identifizierte Person erteilt dann ihrem Benutzer die Erlaubnis, die Rolle zu übernehmen. Der Benutzer kann dann wie folgt auf Objekte zugreifen:

- Die Rolle einnehmen und daraufhin temporäre Sicherheitsanmeldeinformationen erhalten.
- Verwenden der temporären Sicherheitsanmeldeinformationen, um auf die Objekte im Bucket zuzugreifen.

Weitere Informationen zu IAM [IAMRollen](#) finden Sie im IAMBenutzerhandbuch unter Rollen.

Im Folgenden finden Sie eine Zusammenfassung der exemplarischen Schritte:



1. Der Administrator-Benutzer von Konto A ordnet eine Bucket-Richtlinie zu, die Konto B die bedingte Berechtigung erteilt, Objekte hochzuladen.
2. Der Administrator von Konto A erstellt eine IAM Rolle und stellt so eine Vertrauensstellung mit Konto C her, sodass Benutzer in diesem Konto auf Konto A zugreifen können. Die der Rolle zugeordnete Zugriffsrichtlinie schränkt ein, was der Benutzer in Konto C tun kann, wenn der Benutzer auf Konto A zugreift.
3. Der Administrator von Konto B lädt ein Objekt in den Bucket hoch, der Konto A gehört, und erteilt dem Bucket-Eigentümer vollständige Berechtigungen.

4. Der Administrator von Konto C erstellt einen Benutzer und ordnet ihm eine Benutzerrichtlinie zu, die dem Benutzer gestattet, die Rolle zu übernehmen.
5. Der Benutzer in Konto C übernimmt zuerst die Rolle, womit er temporäre Sicherheitsanmeldeinformationen erhält. Unter Verwendung dieser temporären Sicherheitsanmeldeinformationen greift der Benutzer dann auf die Objekte im Bucket zu.

Für dieses Beispiel benötigen Sie drei Konten. Die folgende Tabelle zeigt, wie wir auf diese Konten und die Administrator-Benutzer in diesen Konten verweisen. Gemäß den IAM Richtlinien (siehe [Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen](#)) verwenden wir nicht Root-Benutzer des AWS-Kontos Anmeldeinformationen in dieser exemplarischen Vorgehensweise. Stattdessen erstellen Sie einen Administrator-Benutzer in jedem Konto, und verwenden dessen Anmeldeinformationen, um Ressourcen zu erstellen und ihnen Berechtigungen zu erteilen.

AWS-Konto ID	Konto bezeichnet als	Administratorbenutzer im Konto
<i>1111-1111-1111</i>	Konto A	AccountAdmin
<i>2222-2222-2222</i>	Konto B	AccountBAdmin
<i>3333-3333-3333</i>	Konto C	AccountCAdmin

Schritt 0: Vorbereitung auf den Walkthrough

Note

Möglicherweise möchten Sie einen Texteditor öffnen und einige Informationen aufschreiben, während Sie die Schritte ausführen. Insbesondere benötigen Sie ein KontoIDs, einen kanonischen BenutzerIDs, eine IAM Benutzeranmeldung URLs für jedes Konto, um eine Verbindung zur Konsole herzustellen, sowie Amazon-Ressourcennamen (ARNs) der IAM Benutzer und Rollen.


1. Stellen Sie sicher, dass Sie drei haben AWS-Konten und jedes Konto hat einen Administratorbenutzer, wie in der Tabelle im vorherigen Abschnitt dargestellt.

- a. Melden Sie sich an für AWS-Konten, nach Bedarf. Wir bezeichnen diese Konten als Konto A, Konto B und Konto C.
 - b. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der [IAMKonsole](#) an und gehen Sie wie folgt vor, um einen Administratorbenutzer zu erstellen:
 - Erstellen Sie einen Benutzer **AccountAdmin** und notieren Sie sich seine Sicherheitsanmeldedaten. Weitere Informationen zum Hinzufügen von Benutzern finden Sie unter [Einen IAM Benutzer erstellen in Ihrem AWS-Konto](#) im IAM-Benutzerhandbuch.
 - Gewähren Sie Administratorrechte, AccountAdmin indem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt. Anweisungen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinien verwalten](#).
 - Notieren Sie sich im IAM Konsolen-Dashboard die IAMBenutzeranmeldung URL. Benutzer in diesem Konto müssen dies verwendenURL, wenn sie sich bei der AWS Management Console. Weitere Informationen finden Sie unter [Anmelden bei AWS Management Console als IAM Benutzer](#) im IAMBenutzerhandbuch.
 - c. Wiederholen Sie den obigen Schritt, um Administrator-Benutzer in Konto B und Konto C zu erstellen.
2. Notieren Sie sich für Konto C die kanonische Benutzer-ID.

Wenn Sie eine IAM Rolle in Konto A erstellen, erteilt die Vertrauensrichtlinie Konto C die Erlaubnis, die Rolle anzunehmen, indem die Konto-ID angegeben wird. Sie finden die Konteninformationen wie folgt:

- a. Verwenden Sie Ihre AWS-Konto ID oder Kontoalias, Ihr IAM Benutzername und Ihr Passwort für die Anmeldung an der [Amazon S3 S3-Konsole](#).
 - b. Wählen Sie den Namen eines Amazon-S3-Buckets aus, um die Details zu diesem Bucket anzuzeigen.
 - c. Klicken Sie auf den Tab Berechtigungen und anschließend auf Access Control List.
 - d. Im Access für Ihr AWS-Kontolm Abschnitt Konto befindet sich in der Spalte Konto eine lange Kennung, z.
c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6 B.
Dies ist Ihre kanonische Benutzer-ID.
3. Für das Erstellen einer Bucket-Richtlinie benötigen Sie die folgenden Informationen. Beachten Sie diese Werte:

- Kanonische Benutzer-ID von Konto A – Wenn der Administrator von Konto A dem Administrator von Konto B die bedingte Berechtigung erteilt, Objekte hochzuladen, gibt die Bedingung die kanonische Benutzer-ID des Benutzers von Konto A an, der vollständige Kontrolle über die Objekte benötigt.

 Note

Die kanonische Benutzer-ID ist ein nur für Amazon S3 geltendes Konzept. Es handelt sich dabei um eine 64 Zeichen lange verschleierte Version der Konto-ID.

- Benutzer ARN für Konto B-Administrator — Sie finden den Benutzer ARN in der [IAMKonsole](#). Sie müssen den Benutzer auswählen und den Benutzer auf der Registerkarte „Übersicht“ suchen. ARN

In der Bucket-Richtlinie erteilen Sie die AccountAdmin Berechtigung zum Hochladen von Objekten und geben den Benutzer an, der den ARN verwendet. Hier ist ein ARN Beispielwert:

```
arn:aws:iam::AccountB-ID:user/AccountAdmin
```

4. Richten Sie entweder AWS Command Line Interface (CLI) oder das AWS Tools for Windows PowerShell. Stellen Sie sicher, dass Sie die Administrator-Benutzeranmeldedaten wie folgt speichern:
 - Wenn Sie das verwenden AWS CLI, erstellen Sie Profile AccountAdmin und AccountAdmin, in der Konfigurationsdatei.
 - Wenn Sie die verwenden AWS Tools for Windows PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als AccountAdmin und speichernAccountAdmin.

Detaillierte Anweisungen finden Sie unter [Einrichtung der Tools für die exemplarischen Vorgehensweisen](#).

Schritt 1: Erledigen der Aufgaben von Konto A

In diesem Beispiel ist Konto A der Bucket-Eigentümer. Der Benutzer AccountAdmin in Konto A wird also Folgendes tun:

- Erstellen Sie einen Bucket.

- Hängen Sie eine Bucket-Richtlinie an, die dem Konto B Administratorrechte zum Hochladen von Objekten gewährt.
- Erstellen Sie eine IAM Rolle, die Konto C die Berechtigung erteilt, die Rolle zu übernehmen, sodass es auf Objekte im Bucket zugreifen kann.

Schritt 1.1: Melden Sie sich an bei AWS Management Console

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto A zunächst bei AWS Management Console als **AccountAdmin** Benutzer. Dieser Benutzer erstellt einen Bucket und ordnet ihm eine Richtlinie zu.

Schritt 1.2: Erstellen eines Buckets und Anfügen einer Richtlinie

Führen Sie in der Amazon-S3-Konsole die folgenden Schritte aus:

1. Erstellen Sie einen Bucket. Diese Übung setzt voraus, dass der Bucket-Name *amzn-s3-demo-bucket1* ist.

Detaillierte Anweisungen finden Sie unter [Erstellen eines Buckets](#).

2. Hängen Sie die folgende Bucket-Richtlinie an. Die Richtlinie gewährt dem Administrator für Konto B bedingte Berechtigungen zum Hochladen von Objekten.

Aktualisieren Sie die Richtlinie, indem Sie Ihre eigenen Werte für *amzn-s3-demo-bucket1AccountB-ID*, und die angeben *CanonicalUserId-of-AWSaccountA-BucketOwner*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "111",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::amzn-s3-demo-bucket1/*"
    },
    {
      "Sid": "112",
```

```

    "Effect": "Deny",
    "Principal": {
      "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3::amzn-s3-demo-bucket1/*",
    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-
AWSaccountA-BucketOwner"
      }
    }
  ]
}

```

Schritt 1.3: Erstellen Sie eine IAM Rolle, um Konto C den kontoübergreifenden Zugriff auf Konto A zu ermöglichen

Erstellen Sie in der [IAMKonsole](#) eine IAM Rolle (**examplerole**), die Konto C die Erlaubnis erteilt, die Rolle zu übernehmen. Stellen Sie sicher, dass Sie immer noch als Administrator für Konto A angemeldet sind, da die Rolle in Konto A erstellt werden muss.

1. Bevor Sie die Rolle erstellen, richten Sie die verwaltete Richtlinie ein, die die von der Rolle benötigten Berechtigungen definiert. Diese Richtlinie fügen Sie zu einem späteren Zeitpunkt der Rolle an.
 - a. Wählen Sie im Navigationsbereich auf der linken Seite Richtlinien und dann Richtlinie erstellen aus.
 - b. Klicken Sie neben Create Your Own Policy auf Select.
 - c. Geben Sie **access-accountA-bucket** in das Feld Policy Name ein.
 - d. Kopieren Sie die folgende Zugriffsrichtlinie und fügen Sie sie in das Feld Policy Document ein. Die Zugriffsrichtlinie gewährt der Rolle die `s3:GetObject` Berechtigung, sodass der Benutzer des Kontos C, wenn er die Rolle annimmt, nur den `s3:GetObject` Vorgang ausführen kann.

```

{
  "Version": "2012-10-17",
  "Statement": [

```

```
{
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
}
]
```

- e. Wählen Sie Create Policy (Richtlinie erstellen) aus.

Die neue Richtlinie wird in der Liste der verwalteten Richtlinien angezeigt.

2. Wählen Sie im Navigationsbereich auf der linken Seite Rollen und dann Neue Rolle erstellen aus.
3. Wählen Sie unter Rollentyp auswählen die Option Rolle für kontoübergreifenden Zugriff aus und klicken Sie dann auf die Schaltfläche Auswählen neben Zugriff bereitstellen zwischen AWS-Konten Ihnen gehört.
4. Geben Sie die Konto-ID von Konto C ein.

Für diese exemplarische Vorgehensweise müssen Sie nicht verlangen, dass Benutzer über eine Multi-Faktor-Authentifizierung (MFA) verfügen, um die Rolle übernehmen zu können. Lassen Sie diese Option daher unausgewählt.

5. Klicken Sie auf Next Step, um die mit der Rolle verknüpften Berechtigungen einzurichten.
6. Aktivieren Sie das Kontrollkästchen neben der Access-Account-A-Bucket-Richtlinie, die Sie erstellt haben, und wählen Sie dann Next Step aus.

Die Prüfseite wird angezeigt, sodass Sie die Einstellungen bestätigen können, bevor Sie die Rolle erstellen. Ein sehr wichtiges, auf dieser Seite zu beachtendes Element ist der Link, den Sie an die Benutzer senden können, die die Rolle verwenden müssen. Benutzer, die den Link verwenden, gelangen direkt zur Seite „Rolle wechseln“, auf der die Felder Konto-ID und Rollenname bereits ausgefüllt sind. Dieser Link wird auch auf der Seite Role Summary für beliebige kontoübergreifende Rollen angezeigt.

7. Geben Sie `examplerole` den Rollennamen ein und wählen Sie dann Next Step aus.
8. Nachdem Sie die Rolle überprüft haben, wählen Sie Create Role aus.

Die Rolle `examplerole` wird in der Liste der Rollen angezeigt.

9. Wählen Sie den Rollennamen `examplerole`.
10. Wählen Sie den Tab Trust Relationships.

11. Wählen Sie Richtliniendokument anzeigen und überprüfen Sie, ob die angezeigte Vertrauensrichtlinie mit der folgenden Richtlinie übereinstimmt.

Die folgende Vertrauensrichtlinie richtet eine Vertrauensbeziehung zu Konto C ein und gestattet ihm die Aktion `sts:AssumeRole`. Weitere Informationen finden Sie unter [AssumeRole](#) in der AWS Security Token Service APIReferenz.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountC-ID:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

12. Notieren Sie sich den Amazon-Ressourcennamen (ARN) der `exampleRole` Rolle, die Sie erstellt haben.

Später in den folgenden Schritten fügen Sie eine Benutzerrichtlinie hinzu, die es einem IAM Benutzer ermöglicht, diese Rolle anzunehmen, und Sie identifizieren die Rolle anhand des ARN Werts.

Schritt 2: Erledigen der Aufgaben von Konto B

Der Beispiel-Bucket, der Konto A gehört, benötigt Objekte, die anderen Konten gehören. In diesem Schritt lädt der Administrator von Konto B unter Verwendung der Befehlszeilen-Tools ein Objekt hoch.

- Verwendung der `put-object` AWS CLI Befehl, laden Sie ein Objekt hoch in *amzn-s3-demo-bucket1*.

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --
body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucketOwner" --
profile AccountAdmin
```

Beachten Sie Folgendes:

- Der `--Profile` Parameter gibt das `AccountAdmin` Profil an, sodass das Objekt dem Konto B gehört.
- Der Parameter `grant-full-control` erteilt dem Bucket-Eigentümer vollständige Berechtigungen für das Objekt, wie für die Bucket-Richtlinie erforderlich.
- Der Parameter `--body` identifiziert die hochzuladende Quelldatei. Wenn sich die Datei beispielsweise auf dem Laufwerk C: eines Windows Computer, den Sie angeben: `\HappyFace.jpg`.

Schritt 3: Führen Sie die Aufgaben von Konto C aus

In den vorherigen Schritten hat Konto A bereits eine Rolle `exampleRole`, die Vertrauen zu Konto C herstellt. Diese Rolle ermöglicht es Benutzern in Konto C, auf Konto A zuzugreifen. In diesem Schritt erstellt der Administrator von Konto C einen Benutzer (Dave) und delegiert ihm die von Konto A erhaltenen `sts:AssumeRole` Berechtigungen. Dieser Ansatz ermöglicht es Dave, das Konto A zu übernehmen `exampleRole` und vorübergehend Zugriff auf Konto A zu erhalten. Die Zugriffsrichtlinie, die Konto A der Rolle beigefügt hat, schränkt ein, was Dave tun kann, wenn er auf Konto A zugreift, insbesondere Objekte in `amzn-s3-demo-bucket1`.

Schritt 3.1: Erstellen Sie einen Benutzer in Konto C und delegieren Sie die Zugriffsrechte `exampleRole`

1. Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto C zunächst bei AWS Management Console als **AccountAdmin** Benutzer.
2. Erstellen Sie in der [IAMKonsole](#) einen Benutzer, Dave.

step-by-step Anweisungen finden Sie unter IAM Benutzer [erstellen \(AWS Management Console\)](#) im IAMBenutzerhandbuch.

3. Notieren Sie sich die Dave-Anmeldeinformationen. Dave benötigt diese Anmeldeinformationen, um die Rolle `exampleRole` zu übernehmen.
4. Erstellen Sie eine Inline-Richtlinie für den IAM Dave-Benutzer, um Dave die `sts:AssumeRole` Berechtigung für die `exampleRole` Rolle in Konto A zu delegieren.
 - a. Wählen Sie im Navigationsbereich auf der linken Seite `Users (Benutzer)`.
 - b. Wählen Sie den Benutzernamen Dave.

- c. Wählen Sie auf der Seite mit den Benutzerinformationen den Tab Permissions aus und erweitern Sie den Abschnitt Inline Policies.
- d. Wählen Sie hier klicken (oder Create User Policy).
- e. Wählen Sie Custom Policy und dann Select aus.
- f. Geben Sie einen Namen für die Richtlinie in das Feld Policy Name ein.
- g. Kopieren Sie die folgende Richtlinie in das Feld Policy Document:.

Sie müssen die Richtlinie aktualisieren, indem Sie die angeben *AccountA-ID*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["sts:AssumeRole"],
      "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"
    }
  ]
}
```

- h. Klicken Sie auf Apply Policy (Richtlinie anwenden).
5. Speichern Sie Daves Anmeldeinformationen in der Konfigurationsdatei von AWS CLI durch Hinzufügen eines weiteren Profils, AccountCDave.

```
[profile AccountCDave]
aws_access_key_id = UserDaveAccessKeyID
aws_secret_access_key = UserDaveSecretAccessKey
region = us-west-2
```

Schritt 3.2: Rolle übernehmen (examplerole) und auf Objekte zugreifen

Jetzt kann Dave auf Objekte in dem Bucket zugreifen, der Konto A gehört, nämlich wie folgt:

- Dave übernimmt zuerst die *examplerole* unter Verwendung seiner eigenen Anmeldeinformationen. Damit werden temporäre Anmeldeinformationen zurückgegeben.
- Unter Verwendung der temporären Anmeldeinformationen greift Dave dann auf die Objekte im Bucket von Konto A zu.

1. Führen Sie in der Befehlszeile den folgenden Befehl aus AWS CLI `assume-role` Befehl unter Verwendung des `AccountCDave` Profils.

Sie müssen den ARN Wert im Befehl aktualisieren, indem Sie angeben *AccountA-ID*, wo definiert `examplerole` ist.

```
aws sts assume-role --role-arn arn:aws:iam::AccountA-ID:role/examplerole --profile AccountCDave --role-session-name test
```

Als Antwort AWS Security Token Service (AWS STS) gibt temporäre Sicherheitsanmeldedaten zurück (Zugriffsschlüssel-ID, geheimer Zugriffsschlüssel und ein Sitzungstoken).

2. Speichern Sie die temporären Sicherheitsanmeldedaten im AWS CLI Konfigurationsdatei unter dem `TempCred` Profil.

```
[profile TempCred]
aws_access_key_id = temp-access-key-ID
aws_secret_access_key = temp-secret-access-key
aws_session_token = session-token
region = us-west-2
```

3. Führen Sie in der Befehlszeile Folgendes aus AWS CLI Befehl für den Zugriff auf Objekte mithilfe der temporären Anmeldeinformationen. Der Befehl gibt beispielsweise das Head-Objekt an, API um Objektmetadaten für das `HappyFace.jpg` Objekt abzurufen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg SaveFileAs.jpg --profile TempCred
```

Die `examplerole` zugeordnete Zugriffsrichtlinie erlaubt die Aktionen, deshalb verarbeitet Amazon S3 die Anforderung. Sie können das mit jeder anderen Aktion für jedes andere Objekt im Bucket ausprobieren.

Wenn Sie beispielsweise eine andere Aktion ausprobieren, wird `get-object-acl` Ihnen die Berechtigung verweigert, weil die Rolle diese Aktion nicht erlaubt.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --profile TempCred
```

Wir haben den Benutzer Dave verwendet, um die Rolle zu übernehmen und unter Verwendung temporärer Anmeldeinformationen auf das Objekt zuzugreifen. Es könnte auch eine Anwendung

in Konto C sein, die auf Objekte in *amzn-s3-demo-bucket1* zugreift. Die Anwendung kann temporäre Anmeldeinformationen erhalten, und Konto C kann die Berechtigung der Anwendung delegieren, um `exampleRole` zu übernehmen.

Schritt 4: Bereinigen

1. Nachdem Sie mit dem Testen fertig sind, können Sie wie folgt aufräumen:
 - Melden Sie sich an bei [AWS Management Console](#) verwenden Sie die Anmeldeinformationen von Konto A und gehen Sie wie folgt vor:
 - Entfernen Sie in der Amazon-S3-Konsole die an *amzn-s3-demo-bucket1* angefügte Bucket-Richtlinie. Löschen Sie in den Bucket Properties die Richtlinie im Abschnitt Permissions.
 - Wenn der Bucket für diese Übung erstellt wurde, löschen Sie in der Amazon-S3-Konsole die Objekte und dann den Bucket.
 - Entfernen Sie in der [IAMKonsole](#) die Daten, die `exampleRole` Sie in Konto A erstellt haben. [Eine step-by-step Anleitung dazu finden Sie im IAM Benutzerhandbuch unter Löschen eines IAM Benutzers](#).
 - Entfernen Sie den AccountAdminBenutzer in der [IAMKonsole](#).
2. Melden Sie sich mit den Anmeldeinformationen für Konto B bei der [IAMKonsole](#) an. Löschen Sie den Benutzer AccountBadmin.
3. Melden Sie sich mit den Anmeldeinformationen von Konto C bei der [IAMKonsole](#) an. Delete AccountCadminund der Benutzer Dave.

Zugehörige Ressourcen

Weitere Informationen zu dieser exemplarischen Vorgehensweise finden Sie in den folgenden Ressourcen im IAMBenutzerhandbuch:

- [Eine Rolle erstellen, um Berechtigungen an einen Benutzer zu delegieren IAM](#)
- [Tutorial: Zugriff auf Across delegieren AWS-Konten Rollen verwenden IAM](#)
- [IAMRichtlinien verwalten](#)

Wie Amazon S3 eine Anforderung autorisiert

Wenn Amazon S3 eine Anforderung erhält – z. B. ein Bucket oder eine Objektoperation –, überprüft es zuerst, ob der Auftraggeber die erforderlichen Berechtigungen besitzt. Amazon S3 bewertet alle relevanten Zugriffsrichtlinien, Benutzerrichtlinien und ressourcenbasierten Richtlinien (Bucket-Richtlinie, Bucket-Zugriffskontrollliste (ACL) und ObjektACL), um zu entscheiden, ob die Anfrage autorisiert werden soll.

Note

Wenn bei der Amazon S3-Berechtigungsprüfung keine gültigen Berechtigungen gefunden werden, wird die Fehlermeldung Zugriff verweigert (403 Forbidden) Permission denied (403 Forbidden) zurückgegeben. Weitere Informationen finden Sie unter [Problembehandlung bei Access Denied \(403 Forbidden\) in Amazon S3](#).

Um festzustellen, ob der Anforderer berechtigt ist, den bestimmten Vorgang auszuführen, geht Amazon S3 bei Eingang einer Anfrage in der angegebenen Reihenfolge wie folgt vor:

1. Konvertiert alle relevanten Zugriffsrichtlinien (Benutzerrichtlinie, Bucket-Richtlinie und ACLs) zur Laufzeit in eine Reihe von Richtlinien zur Auswertung.
2. Es wertet in den folgenden Schritten die resultierende Richtlinienmenge aus. Amazon S3 wertet in jedem Schritt eine Untermenge der Richtlinien in einem spezifischen Kontext aus, basierend auf der Kontextautorität.
 - a. Benutzerkontext – Im Benutzerkontext ist das übergeordnete Konto, zu dem der Benutzer gehört, die Kontextautorität.

Amazon S3 wertet eine Untermenge der Richtlinien aus, die dem übergeordneten Konto gehören. Diese Untermenge beinhaltet die Benutzerrichtlinie, die das übergeordnete Konto dem Benutzer zuordnet. Wenn das übergeordnete Objekt auch Eigentümer der Ressource in der Anfrage ist (Bucket oder Objekt), bewertet Amazon S3 gleichzeitig auch die entsprechenden Ressourcenrichtlinien (Bucket-RichtlinieACL, Bucket und ObjektACL).

Ein Benutzer benötigt die Berechtigung von dem übergeordneten Konto, um die Operation auszuführen.

Dieser Schritt gilt nur, wenn die Anfrage von einem Benutzer in einem AWS-Konto. Wenn die Anfrage unter Verwendung der Root-Benutzeranmeldeinformationen eines AWS-Konto, Amazon S3 überspringt diesen Schritt.

- b. Bucket-Kontext — Im Bucket-Kontext bewertet Amazon S3 Richtlinien, die Eigentum von AWS-Konto dem der Bucket gehört.

Erfolgt die Anforderung für eine Bucket-Operation, muss der Auftraggeber die Berechtigung vom Bucket-Eigentümer besitzen. Erfolgt die Anforderung für ein Objekt, wertet Amazon S3 alle Richtlinien aus, die dem Bucket-Eigentümer gehören, um zu überprüfen, ob der Bucket-Eigentümer für das Objekt eine explizite Zugriffsverweigerung festgelegt hat. Wurde eine explizite Zugriffsverweigerung festgelegt, autorisiert Amazon S3 die Anforderung nicht.

- c. Objektkontext – Erfolgt die Anforderung für ein Objekt, wertet Amazon S3 die Untermenge der Richtlinien aus, die dem Objekteigentümer gehören.

Im Folgenden finden Sie einige Beispielszenarien, die veranschaulichen, wie Amazon S3 eine Anfrage autorisiert.

Example — Der Anforderer ist ein Principal IAM

Wenn es sich bei dem Antragsteller um einen IAM Principal handelt, muss Amazon S3 feststellen, ob das übergeordnete Unternehmen AWS-Konto der Auftraggeber, dem der Auftraggeber angehört, hat dem Auftraggeber die erforderliche Genehmigung zur Durchführung des Vorgangs erteilt. Erfolgt die Anforderung darüber hinaus für eine Bucket-Operation, wie beispielsweise eine Anforderung, den Bucket-Inhalt aufzulisten, muss Amazon S3 prüfen, ob der Bucket-Eigentümer dem Auftraggeber die Berechtigung erteilt hat, die Operation auszuführen. Um einen bestimmten Vorgang auf einer Ressource auszuführen, benötigt ein IAM Principal die Erlaubnis der beiden übergeordneten Ressourcen AWS-Konto zu dem es gehört und AWS-Konto dem die Ressource gehört.

Example — Der Anforderer ist ein IAM Principal — Wenn es sich bei der Anfrage um eine Operation an einem Objekt handelt, das dem Bucket-Besitzer nicht gehört

Wenn es sich bei der Anfrage um einen Vorgang mit einem Objekt handelt, das nicht dem Bucket-Besitzer gehört, muss Amazon S3 nicht nur sicherstellen, dass der Anforderer über die Berechtigungen des Objekteigentümers verfügt, sondern auch die Bucket-Richtlinie überprüfen, um sicherzustellen, dass der Bucket-Besitzer nicht die explizite Ablehnung für das Objekt festgelegt hat. Ein Bucket-Eigentümer (der die Rechnung zahlt) kann explizit den Zugriff auf Objekte im Bucket

verweigern, unabhängig davon, wem dieser gehört. Der Bucket-Eigentümer kann auch ein beliebiges Objekt im Bucket löschen.

Standardmäßig, wenn ein anderer AWS-Konto lädt ein Objekt in Ihren S3-Bucket hoch. Dieses Konto (der Objektschreiber) besitzt das Objekt, hat Zugriff darauf und kann anderen Benutzern über Zugriffskontrolllisten (ACLs) Zugriff darauf gewähren. Mit Object Ownership können Sie dieses Standardverhalten so ändern, dass ACLs es deaktiviert ist und Sie als Bucket-Besitzer automatisch Eigentümer aller Objekte in Ihrem Bucket sind. Daher basiert die Zugriffskontrolle für Ihre Daten auf Richtlinien wie IAM Benutzerrichtlinien, S3-Bucket-Richtlinien, Virtual Private Cloud (VPC) - Endpunktrichtlinien und AWS Organizations Richtlinien zur Servicesteuerung (SCPs). Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Weitere Informationen darüber, wie Amazon S3 Zugriffsrichtlinien zur Autorisierung oder Ablehnung von Anfragen für Bucket-Vorgänge und Objekt-Vorgänge bewertet, finden Sie in den folgenden Themen:

Themen

- [Wie Amazon S3 eine Anforderung für eine Bucket-Operation autorisiert](#)
- [Wie Amazon S3 eine Anforderung für eine Objekt-Operation autorisiert](#)

Wie Amazon S3 eine Anforderung für eine Bucket-Operation autorisiert

Wenn Amazon S3 eine Anfrage für eine Bucket-Operation erhält, wandelt Amazon S3 alle relevanten Berechtigungen in eine Reihe von Richtlinien um, die zur Laufzeit ausgewertet werden sollen. Zu den relevanten Berechtigungen gehören ressourcenbasierte Berechtigungen (z. B. Bucket-Richtlinien und Bucket-Zugriffskontrolllisten) und Benutzerrichtlinien, falls die Anfrage von einem IAM Principal stammt. Amazon S3 bewertet dann den resultierenden Satz von Richtlinien in einer Reihe von Schritten entsprechend einem bestimmten Kontext — Benutzerkontext oder Bucket-Kontext:

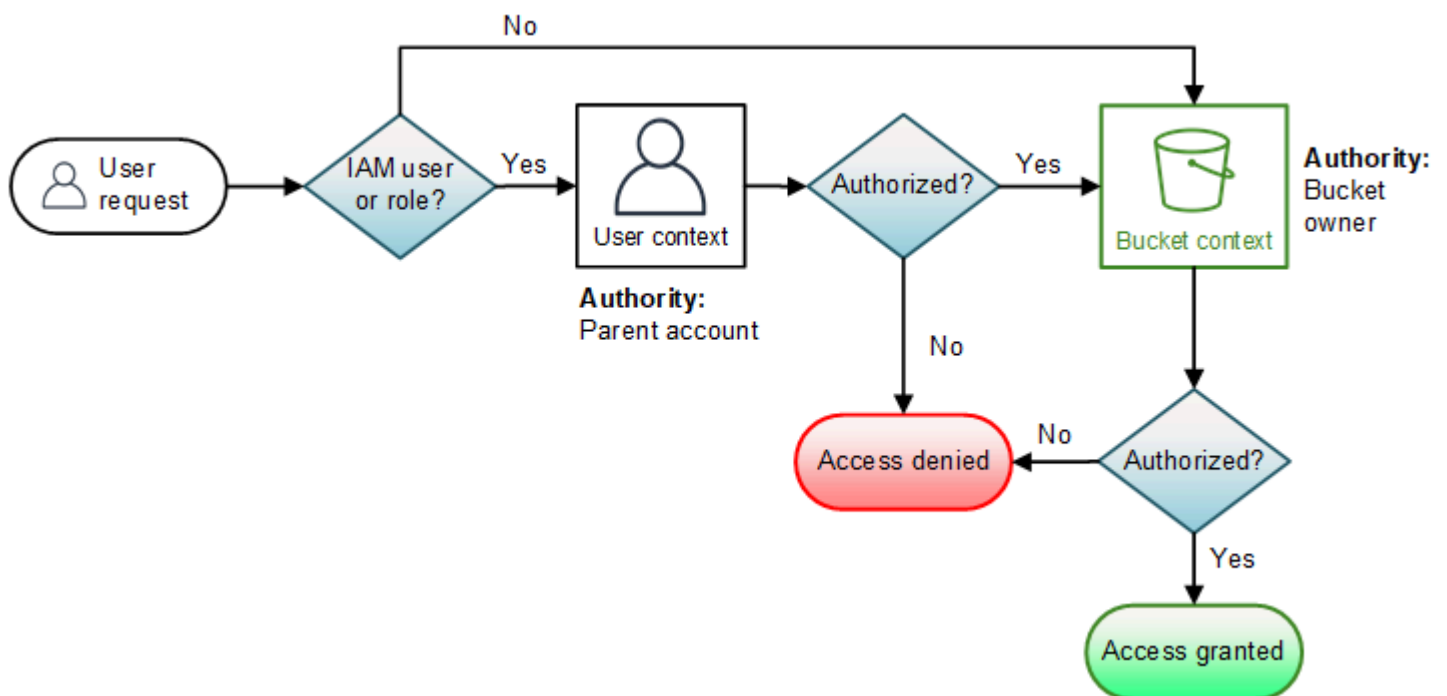
1. Benutzerkontext — Wenn der Anforderer ein IAM Principal ist, muss der Principal die Erlaubnis des übergeordneten Unternehmens haben AWS-Konto zu dem es gehört. In diesem Schritt wertet Amazon S3 eine Untermenge der Richtlinien aus, die dem übergeordneten Konto gehören (auch als Kontextautorität bezeichnet). Diese Richtlinienuntermenge beinhaltet die Benutzerrichtlinie, die das übergeordnete Konto dem Prinzipal zuordnet. Wenn das Elternteil auch Eigentümer der Ressource in der Anfrage ist (in diesem Fall der Bucket), bewertet Amazon S3 gleichzeitig auch die entsprechenden Ressourcenrichtlinien (Bucket-Richtlinie und BucketACL). Immer wenn eine Anforderung für eine Bucket-Operation gemacht wird, zeichnen die Server-Zugriffsprotokolle die

kanonische Benutzer-ID des Anforderers auf. Weitere Informationen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

2. Bucket-Kontext – Der Anforderer muss die Berechtigung vom Bucket-Eigentümer besitzen, eine spezifische Bucket-Operation auszuführen. In diesem Schritt bewertet Amazon S3 eine Teilmenge von Richtlinien, deren Eigentümer AWS-Konto dem der Bucket gehört.

Der Bucket-Besitzer kann die Erlaubnis mithilfe einer Bucket-Richtlinie oder eines Buckets erteilen ACL. Wenn das Symbol AWS-Konto dem der Bucket gehört, ist auch das übergeordnete Konto eines IAM Prinzipals. Anschließend kann er Bucket-Berechtigungen in einer Benutzerrichtlinie konfigurieren.

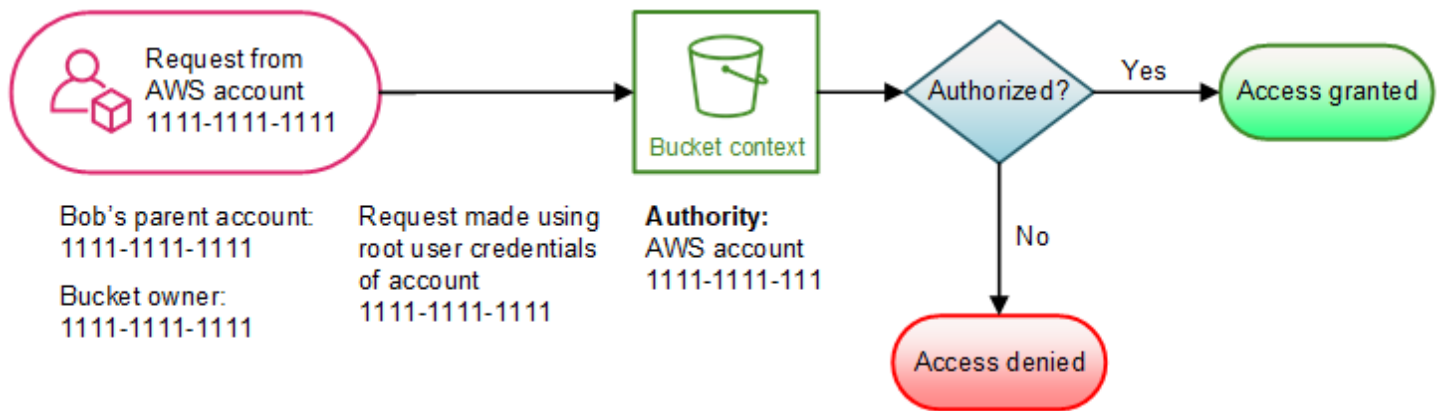
Nachfolgend sehen Sie eine grafische Darstellung der kontextbasierten Auswertung für Bucket-Operationen.



Die folgenden Beispiele veranschaulichen die Auswertungslogik.

Beispiel 1: Vom Bucket-Eigentümer angeforderte Bucket-Operation

In diesem Beispiel sendet der Bucket-Besitzer eine Anfrage für einen Bucket-Vorgang unter Verwendung der Root-Anmeldeinformationen des AWS-Konto.

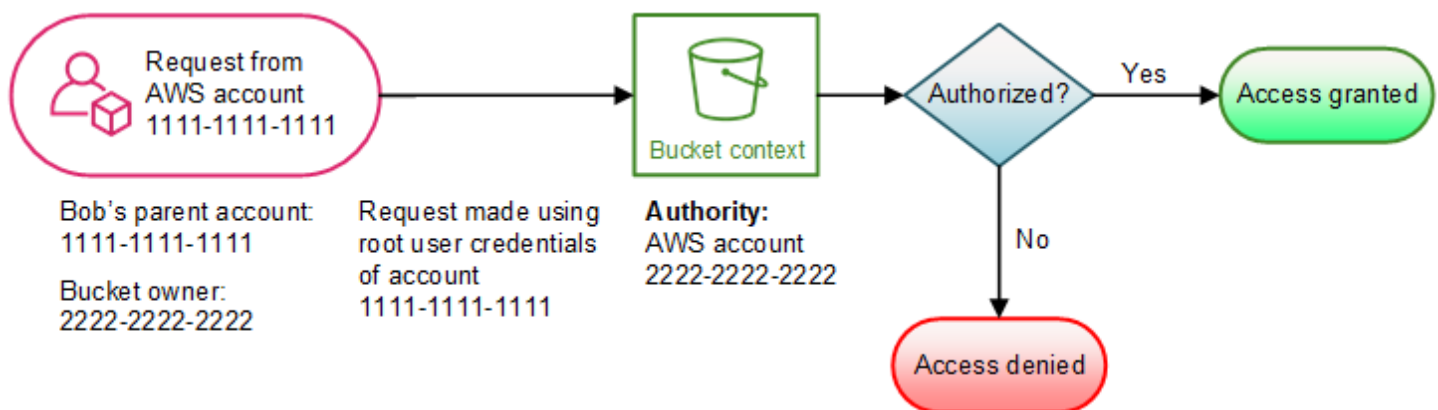


Amazon S3 führt die Kontextauswertung wie folgt durch:

1. Weil die Anfrage unter Verwendung der Root-Benutzeranmeldeinformationen eines AWS-Konto, der Benutzerkontext wird nicht ausgewertet.
2. Im Bucket-Kontext überprüft Amazon S3 die Bucket-Richtlinie, um festzustellen, ob der Auftraggeber die Berechtigung besitzt, die Operation auszuführen. Amazon S3 autorisiert die Anforderung.

Beispiel 2: Bucket-Operation, angefordert von einem AWS-Konto das ist nicht der Bucket-Besitzer

In diesem Beispiel wird eine Anfrage mithilfe der Root-Benutzeranmeldedaten von gestellt AWS-Konto 1111-1111-1111 für einen Bucket-Vorgang, der gehört AWS-Konto 2222-2222-2222. An dieser Anfrage sind keine IAM Benutzer beteiligt.



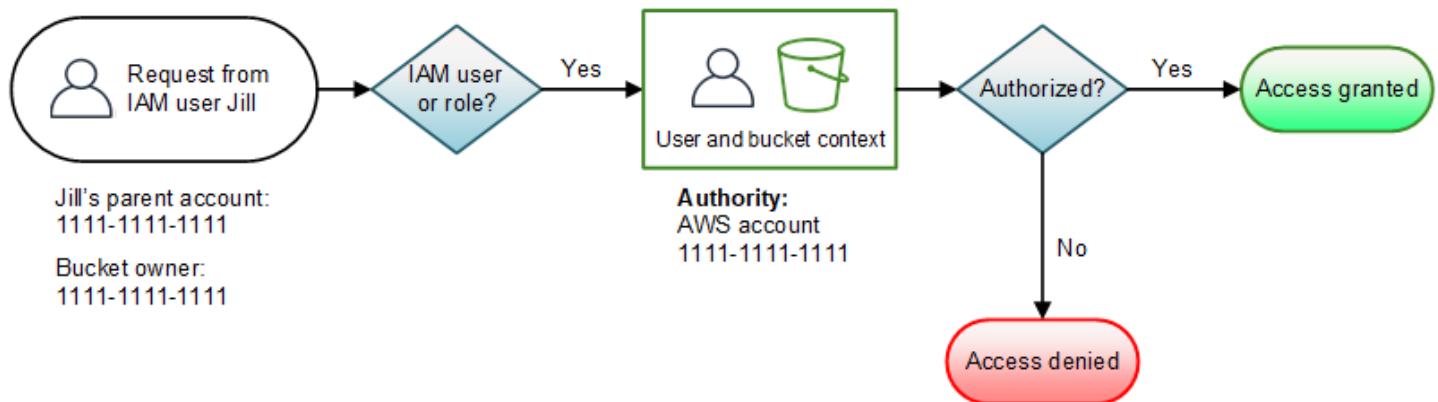
In diesem Beispiel bewertet Amazon S3 den Kontext wie folgt:

1. Weil die Anfrage unter Verwendung der Root-Benutzeranmeldeinformationen eines AWS-Konto, der Benutzerkontext wird nicht ausgewertet.

- Im Bucket-Kontext wertet Amazon S3 die Bucket-Richtlinie aus. Wenn der Bucket-Besitzer (AWS-Konto 2222-2222-2222) hat nicht autorisiert AWS-Konto 1111-1111-1111 Um den angeforderten Vorgang durchzuführen, lehnt Amazon S3 die Anfrage ab. Andernfalls genehmigt Amazon S3 die Anforderung und führt die Operation aus.

Beispiel 3: Bucket-Vorgang, der von einem Principal angefordert wurde, dessen übergeordnetes Element IAM AWS-Konto ist auch der Bucket-Besitzer

Im Beispiel wird die Anfrage von Jill gesendet, einer IAM Benutzerin in AWS-Konto 1111-1111-1111, der auch der Bucket gehört.



Amazon S3 führt die folgende Kontextauswertung durch:

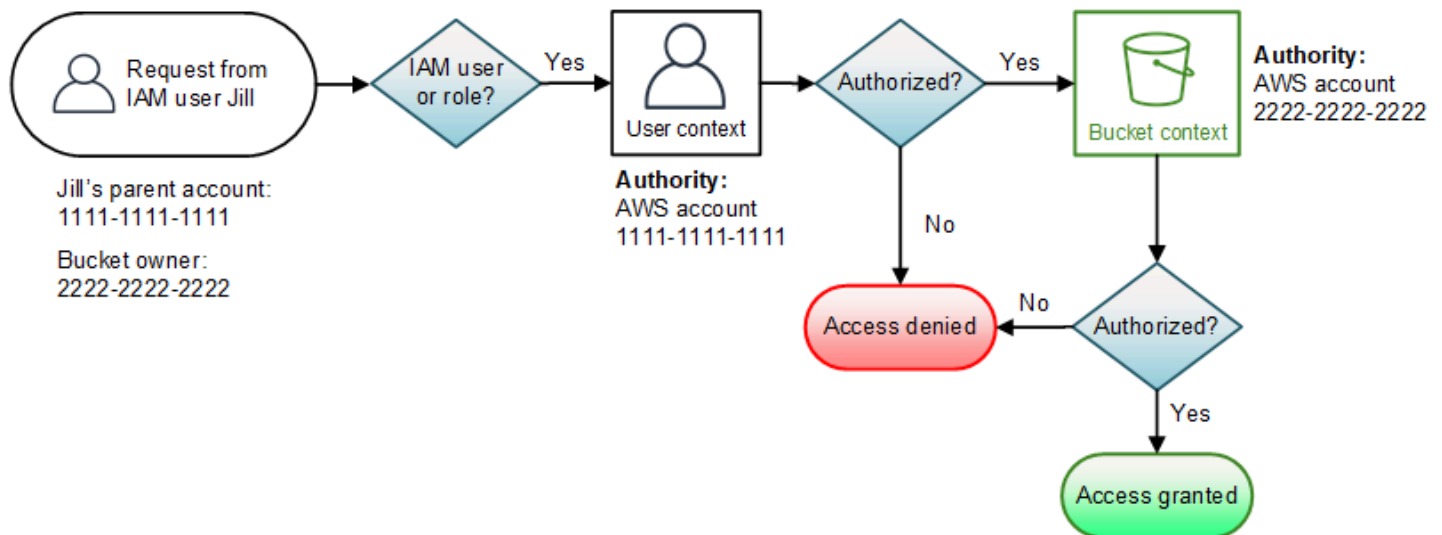
- Da die Anfrage von einem IAM Principal stammt, bewertet Amazon S3 im Benutzerkontext alle Richtlinien, die dem übergeordneten Objekt gehören. AWS-Konto um festzustellen, ob Jill die Erlaubnis hat, den Vorgang durchzuführen.

In diesem Beispiel Elternteil AWS-Konto 1111-1111-1111, zu dem der Principal gehört, ist auch der Bucket-Besitzer. Daher bewertet Amazon S3 zusätzlich zur Benutzerrichtlinie auch die Bucket-Richtlinie und den Bucket ACL im selben Kontext, da sie zu demselben Konto gehören.

- Da Amazon S3 die Bucket-Richtlinie und den Bucket ACL als Teil des Benutzerkontextes bewertet hat, bewertet es den Bucket-Kontext nicht.

Beispiel 4: Bucket-Vorgang, der von einem IAM Principal angefordert wurde, dessen übergeordnetes Element AWS-Konto ist nicht der Bucket-Besitzer

In diesem Beispiel wird die Anfrage von Jill gesendet, einer IAM Benutzerin, deren Elternteil AWS-Konto ist 1111-1111-1111, aber der Bucket gehört einem anderen AWS-Konto, 2222-2222-2222.



Jill benötigt die Berechtigungen beider Elternteile AWS-Konto und der Bucket-Besitzer. Amazon S3 wertet den Kontext wie folgt aus:


1. Da die Anfrage von einem IAM Principal stammt, bewertet Amazon S3 den Benutzerkontext, indem es die vom Konto verfassten Richtlinien überprüft, um sicherzustellen, dass Jill über die erforderlichen Berechtigungen verfügt. Wenn Jill die Erlaubnis hat, bewertet Amazon S3 den Bucket-Kontext weiter. Wenn Jill keine Genehmigung hat, lehnt sie die Anfrage ab.
2. Im Bucket-Kontext überprüft Amazon S3, ob der Bucket-Besitzer 2222-2222-2222 Jill (oder ihrem Elternteil) gewährt hat AWS-Konto) die Erlaubnis, den angeforderten Vorgang durchzuführen. Wenn sie über diese Genehmigung verfügt, erteilt Amazon S3 der Anfrage und führt den Vorgang durch. Andernfalls lehnt Amazon S3 die Anforderung ab.

Wie Amazon S3 eine Anforderung für eine Objekt-Operation autorisiert

Wenn Amazon S3 eine Anfrage für einen Objektvorgang erhält, konvertiert es alle relevanten Berechtigungen — ressourcenbasierte Berechtigungen (Object Access Control List (ACL), Bucket-Richtlinie, BucketACL) und IAM Benutzerrichtlinien — in eine Reihe von Richtlinien, die zur Laufzeit ausgewertet werden. Anschließend wertet es in mehreren Schritten die resultierende Richtlinienmenge aus. In jedem Schritt bewertet es eine Teilmenge von Richtlinien in drei spezifischen Kontexten — Benutzerkontext, Bucket-Kontext und Objektkontext:

1. Benutzerkontext — Wenn der Anforderer ein IAM Principal ist, muss der Principal die Erlaubnis des übergeordneten Unternehmens haben AWS-Konto zu dem es gehört. In diesem Schritt wertet Amazon S3 eine Untermenge der Richtlinien aus, die dem übergeordneten Konto gehören (auch als Kontextautorität bezeichnet). Diese Richtlinienuntermenge beinhaltet die

Benutzerrichtlinie, die das übergeordnete Konto dem Prinzipal zuordnet. Wenn das übergeordnete Objekt auch Eigentümer der Ressource in der Anfrage ist (Bucket oder Objekt), bewertet Amazon S3 gleichzeitig die entsprechenden Ressourcenrichtlinien (Bucket-RichtlinieACL, Bucket und ObjektACL).


 Note

Wenn das Elternteil AWS-Konto besitzt die Ressource (Bucket oder Objekt), kann es seinem IAM Principal Ressourcenberechtigungen gewähren, indem es entweder die Benutzerrichtlinie oder die Ressourcenrichtlinie verwendet.

2. Bucket-Kontext — In diesem Zusammenhang bewertet Amazon S3 Richtlinien, die Eigentum von AWS-Konto dem der Bucket gehört.

Wenn das Symbol AWS-Konto der Eigentümer des Objekts in der Anfrage ist nicht mit dem Bucket-Besitzer identisch. Amazon S3 überprüft die Richtlinien, wenn der Bucket-Besitzer den Zugriff auf das Objekt explizit verweigert hat. Wurde eine explizite Zugriffsverweigerung für das Objekt festgelegt, autorisiert Amazon S3 die Anforderung nicht.

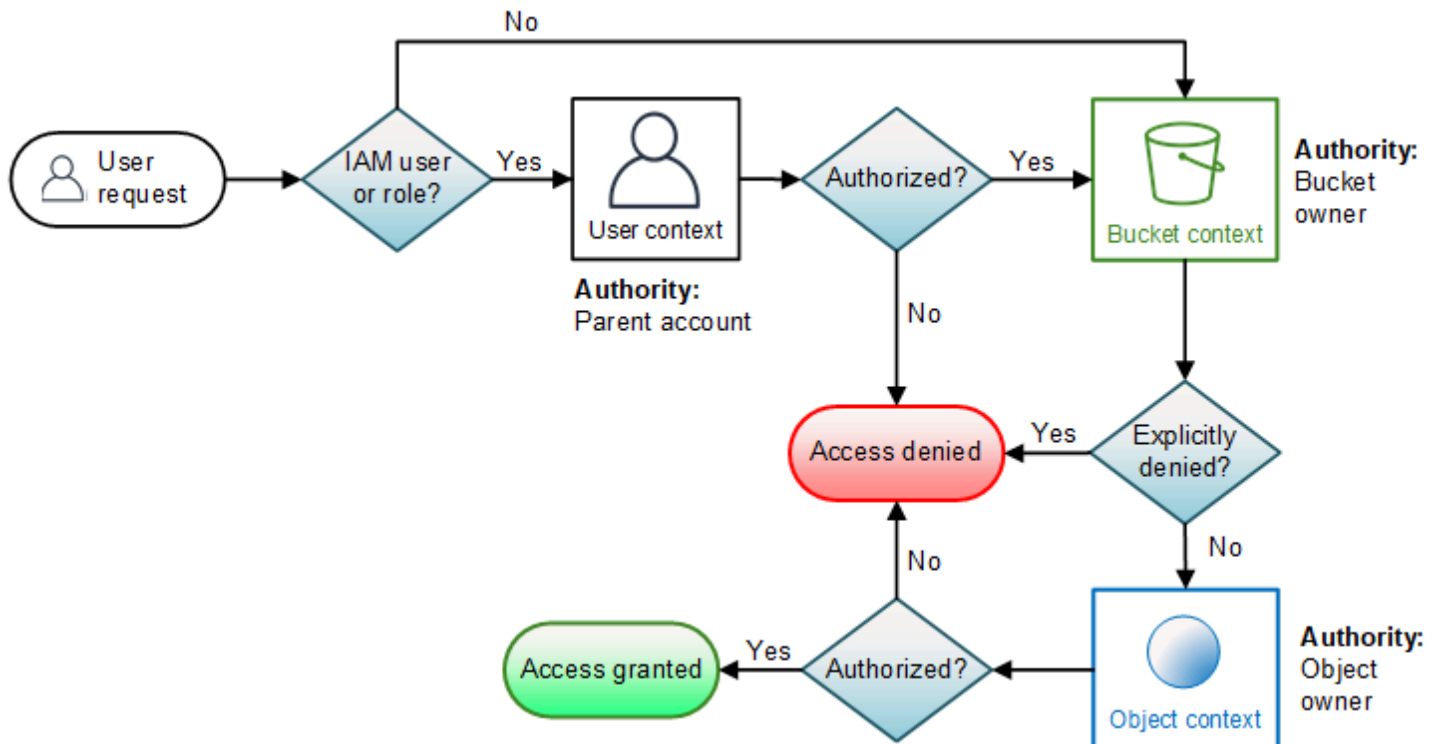
3. Objekt-Kontext – Der Anforderer muss die Berechtigung vom Objekt-Eigentümer besitzen, eine spezifische Objekt-Operation auszuführen. In diesem Schritt bewertet Amazon S3 das ObjektACL.

 Note

Ist der Bucket-Eigentümer gleich dem Objekt-Eigentümer ist, kann der Zugriff auf das Objekt in der Bucket-Richtlinie erteilt werden, die im Bucket-Kontext ausgewertet wird. Wenn die Eigentümer unterschiedlich sind, müssen die Objekteigentümer ein Objekt verwenden, um Berechtigungen ACL zu erteilen. Wenn das Symbol AWS-Konto der Eigentümer des Objekts ist auch das übergeordnete Konto, zu dem der IAM Prinzipal gehört. Es kann Objektberechtigungen in einer Benutzerrichtlinie konfigurieren, die im Benutzerkontext ausgewertet wird. Weitere Informationen über die Verwendung dieser Alternativen zu Zugriffsrichtlinien finden Sie unter [Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten](#). Wenn Sie als Bucket-Besitzer Eigentümer aller Objekte in Ihrem Bucket sein und Bucket-Richtlinien oder -Richtlinien verwenden möchten, die auf der IAMto Verwaltung des Zugriffs auf diese Objekte basieren, können Sie die erzwungene Einstellung des Bucket-Besitzers für Objekteigentümer anwenden. Mit dieser Einstellung besitzen Sie als Bucket-Eigentümer automatisch die volle Kontrolle über jedes Objekt in Ihrem Bucket. Bucket

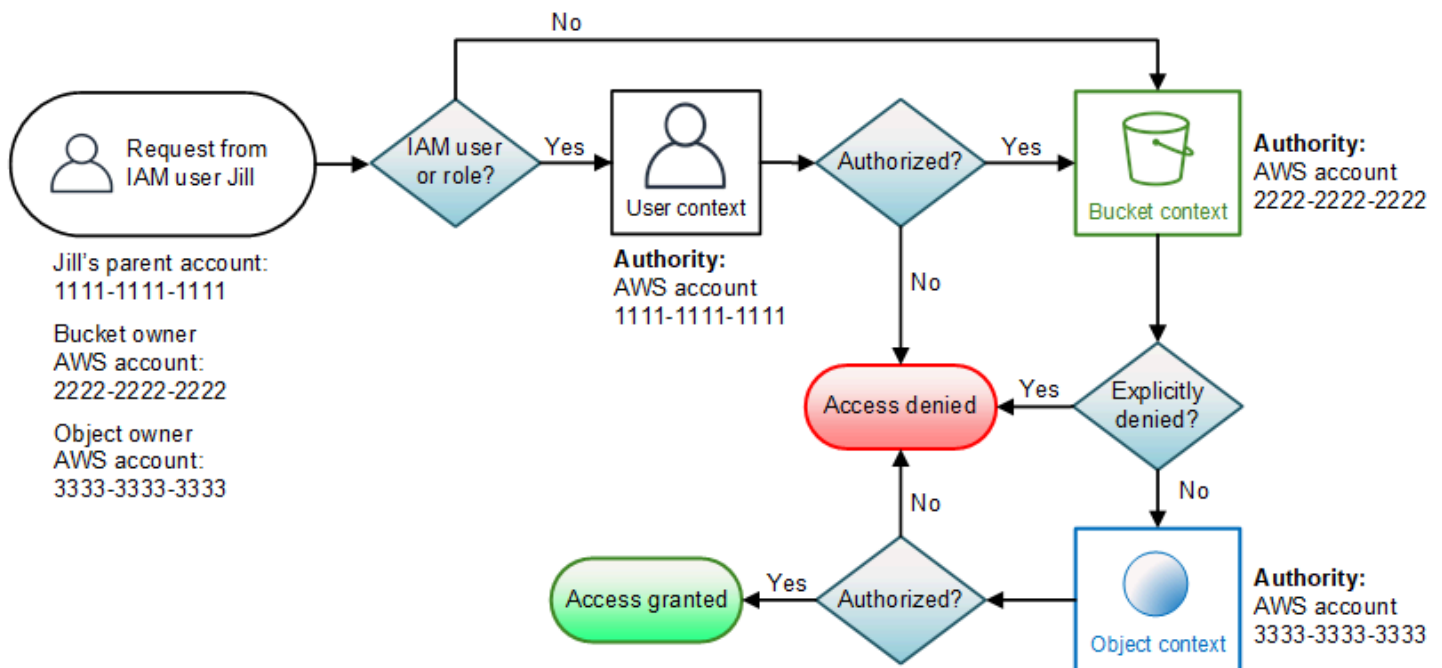
und Objekt ACLs können nicht bearbeitet werden und werden nicht mehr für den Zugriff berücksichtigt. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Nachfolgend sehen Sie eine grafische Darstellung der kontextbasierten Auswertung für eine Objekt-Operation.



Beispiel für eine Anfrage zur Objektoperation

In diesem Beispiel die IAM Benutzerin Jill, deren Elternteil AWS-Konto ist 1111-1111-1111, sendet eine Anforderung zur Objektoperation (z. B.) für ein Objekt, dessen Eigentümer GetObject AWS-Konto 3333-3333-3333 in einem Bucket, der gehört AWS-Konto 2222-2222-2222.



Jill benötigt die Erlaubnis der Eltern AWS-Konto, der Bucket-Besitzer und der Objekteigentümer. Amazon S3 wertet den Kontext wie folgt aus:

1. Da die Anfrage von einem IAM Principal stammt, wertet Amazon S3 den Benutzerkontext aus, um zu überprüfen, ob der übergeordnete AWS-Konto 1111-1111-1111 hat Jill die Erlaubnis erteilt, den angeforderten Vorgang durchzuführen. Wenn sie die Berechtigung besitzt, wertet Amazon S3 den Bucket-Kontext aus. Andernfalls lehnt Amazon S3 die Anforderung ab.
2. Im Bucket-Kontext der Bucket-Besitzer, AWS-Konto 2222-2222-2222 ist die Kontextautorität. Amazon S3 wertet die Bucket-Richtlinie aus, um festzustellen, ob der Bucket-Eigentümer Jill explizit die Berechtigung entzogen hat, auf das Objekt zuzugreifen.
3. Im Objektkontext lautet die Kontextautorität AWS-Konto 3333-3333-3333, der Eigentümer des Objekts. Amazon S3 bewertet das ObjektACL, um festzustellen, ob Jill berechtigt ist, auf das Objekt zuzugreifen. Ist dies der Fall, autorisiert Amazon S3 die Anfrage.

AWS verwaltete Richtlinien für Amazon S3

Importieren in &S3; AWS Eine verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie Folgendes: AWS verwaltete Richtlinien gewähren möglicherweise keine Berechtigungen mit den geringsten Rechten für Ihre spezifischen Anwendungsfälle, da sie für alle verfügbar sind AWS zu verwendende Kunden. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [kundenverwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in definierten Berechtigungen nicht ändern AWS verwaltete Richtlinien. Wenn AWS aktualisiert die in einem AWS bei verwalteter Richtlinie wirkt sich das Update auf alle Hauptidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am wahrscheinlichsten ein AWS verwaltete Richtlinie bei einer neuen AWS-Service wird gestartet oder neue API Operationen werden für bestehende Dienste verfügbar.

Weitere Informationen finden Sie unter [AWS verwaltete Richtlinien](#) im IAM Benutzerhandbuch.

AWS verwaltete Richtlinie: AmazonS3FullAccess

Sie können die AmazonS3FullAccess Richtlinie an Ihre IAM Identitäten anhängen. Diese Richtlinie gewährt Berechtigungen, die vollen Zugriff auf Amazon S3 ermöglichen.

Die Berechtigungen für diese Richtlinie finden Sie unter [AmazonS3FullAccess](#) in der AWS Management Console.

AWS verwaltete Richtlinie: AmazonS3ReadOnlyAccess

Sie können die AmazonS3ReadOnlyAccess Richtlinie an Ihre IAM Identitäten anhängen. Diese Richtlinie gewährt Berechtigungen, die einen schreibgeschützten Zugriff auf Amazon S3 erlauben.

Die Berechtigungen für diese Richtlinie finden Sie unter [AmazonS3ReadOnlyAccess](#) in der AWS Management Console.

AWS verwaltete Richtlinie: AmazonS3ObjectLambdaExecutionRolePolicy

Bietet AWS Lambda verfügt über die erforderlichen Berechtigungen, um Daten an S3 Object Lambda zu senden, wenn Anfragen an einen S3 Object Lambda Access Point gestellt werden. Gewährt Lambda außerdem Berechtigungen zum Schreiben in CloudWatch Amazon-Logs.

Die Berechtigungen für diese Richtlinie finden Sie unter [AmazonS3ObjectLambdaExecutionRolePolicy](#) in der AWS Management Console.

Amazon S3 S3-Updates für AWS Verwaltete Richtlinien

Einzelheiten zu Updates anzeigen für AWS verwaltete Richtlinien für Amazon S3, seit dieser Service begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
Amazon S3 hat die Beschreibungsberechtigungen zu AmazonS3ReadOnlyAccess hinzugefügt.	Amazon S3 hat die s3:Describe* -Berechtigungen zu AmazonS3ReadOnlyAccess hinzugefügt.	11. August 2023
Amazon S3 hat S3 Objekt Lambda-Berechtigungen zu AmazonS3FullAccess und AmazonS3ReadOnlyAccess hinzugefügt	Amazon S3 hat die AmazonS3FullAccess - und AmazonS3ReadOnlyAccess -Richtlinien zur Einbeziehung von Berechtigungen für S3-Objekt-Lambda aktualisiert.	27. September 2021
Amazon S3 AmazonS3ObjectLambdaExecutionRolePolicy hinzugefügt	Amazon S3 hat ein neues hinzugefügt AWS-verwaltete Richtlinie, AmazonS3ObjectLambdaExecutionRolePolicy die Lambda-Funktionen Berechtigungen zur Interaktion mit S3 Object Lambda und zum Schreiben in Protokolle gewährt. CloudWatch	18. August 2021
Amazon S3 hat mit der Verfolgung von Änderungen begonnen	Amazon S3 hat begonnen, Änderungen für seine AWS verwaltete Richtlinien.	18. August 2021

Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens

Um Amazon S3 Storage Lens zu verwenden, um Metriken für all Ihre Konten zu sammeln und zu aggregieren in AWS Organizations, müssen Sie zunächst sicherstellen, dass S3 Storage Lens über einen vertrauenswürdigen Zugriff verfügt, der durch das Verwaltungskonto in Ihrer Organisation aktiviert wurde. S3 Storage Lens erstellt eine mit dem Dienst verknüpfte Rolle (SLR), damit es die

Liste der abrufen kann AWS-Konten gehört zu Ihrer Organisation. Diese Liste von Konten wird von S3 Storage Lens verwendet, um Metriken für S3-Ressourcen in allen Mitgliedskonten zu sammeln, wenn das S3-Storage-Lens-Dashboard oder die Konfigurationen erstellt oder aktualisiert werden.

Verwendungen von Amazon S3 Storage Lens AWS Identity and Access Management (IAM) [Rollen, die mit Diensten verknüpft sind](#). Eine dienstgebundene Rolle ist ein einzigartiger Rollentyp, der IAM direkt mit S3 Storage Lens verknüpft ist. Dienstbezogene Rollen sind von S3 Storage Lens vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere Rollen aufzurufen AWS-Services in Ihrem Namen.

Eine serviceverknüpfte Rolle vereinfacht das Einrichten von S3 Storage Lens, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. S3 Storage Lens definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur S3 Storage Lens die Rollen übernehmen. Zu den definierten Berechtigungen gehören die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen IAM Entität zugeordnet werden.

Sie können diese serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen Ressourcen gelöscht wurden. Dies schützt Ihre S3-Storage-Lens-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Diensten, die dienstbezogene Rollen unterstützen, finden Sie unter [AWS Dienste, die mit Diensten arbeiten IAM](#) und nach Diensten suchen, für die in der Spalte Serviceverknüpfte Rolle der Wert Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für serviceverknüpfte Rollen für Amazon S3 Storage Lens

S3 Storage Lens verwendet die dienstverknüpfte Rolle mit dem Namen `AWSServiceRoleForS3StorageLens`— Dies ermöglicht den Zugriff auf AWS Dienste und Ressourcen, die von S3 Storage Lens verwendet oder verwaltet werden. Dadurch kann S3 Storage Lens darauf zugreifen AWS Organizations Ressourcen in Ihrem Namen.

Die serviceverknüpfte Rolle S3 Storage Lens vertraut dem folgenden Service auf dem Speicher Ihres Unternehmens:

- `storage-lens.s3.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt S3 Storage Lens die Durchführung der folgenden Aktionen:

- `organizations:DescribeOrganization`
`organizations:ListAccounts`
`organizations:ListAWSServiceAccessForOrganization`
`organizations:ListDelegatedAdministrators`

Sie müssen Berechtigungen konfigurieren, damit eine IAM Entität (z. B. ein Benutzer, eine Gruppe oder eine Rolle) eine dienstbezogene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [Berechtigungen für dienstbezogene Rollen](#) im IAMBenutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für S3 Storage Lens

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine der folgenden Aufgaben ausführen, während Sie bei AWS Organizations Für die Verwaltung oder die Administratorkonten für Delegierte erstellt S3 Storage Lens die dienstbezogene Rolle für Sie:

- Erstellen Sie in der Amazon-S3-Konsole eine S3-Storage-Lens-Dashboard-Konfiguration für Ihr Unternehmen.
- PUTeine S3 Storage Lens-Konfiguration für Ihr Unternehmen unter Verwendung von REST API AWS CLI undSDKs.

Note

S3 Storage Lens unterstützt maximal fünf delegierte Administratoren pro Organisation.

Wenn Sie diese serviceverknüpfte Rolle löschen, werden sie von den vorherigen Aktionen bei Bedarf neu erstellt.

Beispielrichtlinie für die serviceverknüpfte Rolle S3 Storage Lens

Example Berechtigungsrichtlinie für die serviceverknüpfte Rolle S3 Storage Lens

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Sid": "AwsOrgsAccess",
    "Effect": "Allow",
    "Action": [
        "organizations:DescribeOrganization",
        "organizations:ListAccounts",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListDelegatedAdministrators"
    ],
    "Resource": [
        "*"
    ]
}
]
```

Bearbeiten einer serviceverknüpften Rolle für Amazon S3 Storage Lens

Mit S3 Storage Lens können Sie das nicht bearbeiten `AWSServiceRoleForS3StorageLens` Rolle, die mit dem Dienst verknüpft ist. Nachdem Sie eine serviceverknüpfte Rolle erstellt haben, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten auf die Rolle verweisen könnten. Sie können die Beschreibung der Rolle jedoch mithilfe IAM von bearbeiten. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Bearbeiten einer dienstbezogenen Rolle](#).

Löschen einer serviceverknüpften Rolle für Amazon S3 Storage Lens

Wenn Sie die serviceverknüpfte Rolle nicht mehr verwenden müssen, empfehlen wir, die Rolle zu löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der Amazon-S3-Storage-Lens-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um das zu löschen `AWSServiceRoleForS3StorageLens` Sie müssen alle S3 Storage Lens-Konfigurationen auf Organisationsebene löschen, die in allen vorhanden sind AWS-Regionen unter Verwendung des AWS Organizations Verwaltungs- oder Delegate-Administratorkonten.

Die Ressourcen sind S3-Storage-Lens-Konfigurationen auf Organisationsebene. Verwenden Sie S3 Storage Lens, um die Ressourcen zu bereinigen, und verwenden Sie dann die [IAMKonsole](#), CLI RESTAPI, oder AWS SDK um die Rolle zu löschen.

In REST API der AWS CLI, und SDKs, S3 Storage Lens-Konfigurationen können `ListStorageLensConfigurations` in allen Regionen ermittelt werden, in denen Ihr Unternehmen S3 Storage Lens-Konfigurationen erstellt hat. Verwenden Sie die Aktion `DeleteStorageLensConfiguration`, um diese Konfigurationen zu löschen, damit Sie die Rolle dann löschen können.

Note

Um die serviceverknüpfte Rolle zu löschen, müssen Sie alle S3-Storage-Lens-Konfigurationen auf Organisationsebene in allen Regionen löschen, in denen sie existieren.

Um Amazon S3 Storage Lens-Ressourcen zu löschen, die verwendet werden von `AWSServiceRoleForS3StorageLens` SLR

1. Um eine Liste Ihrer Konfigurationen auf Organisationsebene zu erhalten, müssen Sie die S3 Storage Lens `ListStorageLensConfigurations` in jeder Region verwenden, in der Sie Konfigurationen von S3 Storage Lens haben. Diese Liste kann auch über die Amazon S3 S3-Konsole abgerufen werden.
2. Löschen Sie diese Konfigurationen von den entsprechenden regionalen Endpunkten, indem `DeleteStorageLensConfiguration` API Sie den Anruf aufrufen oder die Amazon S3 S3-Konsole verwenden.

Um die mit dem Service verknüpfte Rolle manuell zu löschen, verwenden Sie IAM

Nachdem Sie die Konfigurationen gelöscht haben, löschen Sie die `AWSServiceRoleForS3StorageLens` SLR von der [IAMKonsole](#) aus oder durch IAM API `DeleteServiceLinkedRole` Aufrufen von oder mithilfe von AWS CLI or AWS SDK. Weitere Informationen finden Sie im IAM Benutzerhandbuch unter [Löschen einer serviceverknüpften Rolle](#).

Unterstützte Regionen für serviceverknüpfte Rollen mit S3 Storage Lens

S3 Storage Lens unterstützt die Verwendung von serviceverknüpften Rollen in allen AWS-Regionen wo der Dienst verfügbar ist. Weitere Informationen finden Sie unter [Amazon-S3-Regionen und Endpunkte](#).

Problembehandlung Amazon S3 S3-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon S3 und auftreten können IAM.

Themen

- [Ich habe die Fehlermeldung „Zugriff verweigert“ erhalten](#)
- [Ich bin nicht berechtigt, eine Aktion in Amazon S3 durchzuführen](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Leute außerhalb meines AWS-Konto um auf meine Amazon S3 S3-Ressourcen zuzugreifen](#)

Ich habe die Fehlermeldung „Zugriff verweigert“ erhalten

Vergewissern Sie sich, dass weder in der Bucket-Richtlinie noch in der identitätsbasierten Richtlinie eine ausdrückliche Deny Aussage gegen den Antragsteller vorliegt, dem Sie Berechtigungen erteilen möchten.

Ausführliche Informationen zur Behebung von Fehlern bei Zugriffsverweigerung finden Sie unter [Problembehandlung bei Zugriff verweigert \(403 Forbidden\) Fehler in Amazon S3](#)

Ich bin nicht berechtigt, eine Aktion in Amazon S3 durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM Benutzer versucht, die Konsole zu verwenden, um Details zu einer fiktiven `my-example-widget` Ressource anzuzeigen, aber nicht über die fiktiven `s3:GetWidget` Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
s3:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `s3:GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon S3 übergeben können.

Etwas AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon S3 auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Leute außerhalb meines AWS-Konto um auf meine Amazon S3 S3-Ressourcen zuzugreifen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon S3 diese Funktionen unterstützt, finden Sie unter [So funktioniert Amazon S3 mit IAM](#).
- Um zu erfahren, wie Sie Zugriff auf Ihre Ressourcen gewähren können AWS-Konten die Ihnen gehören, finden Sie unter [Gewähren des Zugriffs für einen IAM Benutzer in einem anderen AWS-Konto die Sie besitzen, finden Sie](#) im IAMBenutzerhandbuch.

- Um zu erfahren, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren AWS-Konten, siehe [Zugriff gewähren auf AWS-Konten Eigentum Dritter](#) im IAMBenutzerhandbuch.
- Informationen zur [Bereitstellung des Zugriffs über einen Identitätsverbund finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#). IAM
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM

Verwalten des Zugriffs mit S3-Zugriffsberechtigungen

Zur Einhaltung des Grundsatzes der geringsten Berechtigung definieren Sie die Details des Zugriffs auf Ihre Amazon-S3-Daten basierend auf Anwendungen, Personas, Gruppen oder Organisationseinheiten. Abhängig von Umfang und Komplexität der Zugriffsmuster können Sie verschiedene Ansätze zur Gewährung eines detaillierten Zugriffs auf Ihre Daten in Amazon S3 verwenden.

Der einfachste Ansatz zur Verwaltung des Zugriffs auf small-to-medium eine Reihe von Datensätzen in Amazon S3 von AWS Identity and Access Management (IAM) Das Prinzip besteht darin, [IAMBerechtigungsrichtlinien und S3-Bucket-Richtlinien](#) zu definieren. Diese Strategie funktioniert, solange die erforderlichen Richtlinien innerhalb der Größenbeschränkungen der S3-Bucket-Richtlinien (20 KB) und IAM Richtlinien (5 KB) und innerhalb der [Anzahl der pro Konto zulässigen IAM Prinzipale](#) liegen.

Wenn die Zahl Ihrer Datensätze und Anwendungsfälle wächst, werden auch Richtlinien wichtiger. Ein Ansatz, der Richtlinienanweisungen deutlich mehr Raum gibt, besteht in der Verwendung von [S3-Zugangspunkten](#) als zusätzlichen Endpunkten für S3-Buckets, da jeder Zugangspunkt eine eigene Richtlinie haben kann. Sie können recht detaillierte Zugriffskontrollmuster definieren, da Sie Tausende von Zugriffspunkten pro Person einrichten können AWS-Region pro Konto, mit einer Richtlinie mit einer Größe von bis zu 20 KB für jeden Access Point. S3-Zugangspunkte vergrößern zwar den verfügbaren Raum für Richtlinien, erfordern jedoch einen Mechanismus, mit dem Clients den richtigen Zugangspunkt für den richtigen Datensatz finden können.

Ein dritter Ansatz besteht darin, ein [IAMSitzungsbroker-Muster](#) zu implementieren, bei dem Sie eine Logik für die Zugriffsentscheidung implementieren und für jede Zugriffssitzung dynamisch kurzfristige IAM Sitzungsanmeldeinformationen generieren. Der IAM Sitzungsbroker-Ansatz unterstützt zwar beliebig dynamische Berechtigungsmuster und lässt sich effektiv skalieren, Sie müssen jedoch die Logik für die Zugriffsmuster erstellen.

Sie können anstelle dieser Ansätze S3 Access Grants verwenden, um den Zugriff auf Ihre Amazon-S3-Daten zu verwalten. S3 Access Grants bietet ein vereinfachtes Modell für die Definition von Zugriffsberechtigungen für Daten in Amazon S3 nach Präfix, Bucket oder Objekt. Darüber hinaus können Sie S3 Access Grants verwenden, um sowohl IAM Prinzipalen als auch direkt Benutzern oder Gruppen aus Ihrem Unternehmensverzeichnis Zugriff zu gewähren.

In der Regel definieren Sie Berechtigungen für Daten in Amazon S3, indem Sie Benutzer und Gruppen Datensätzen zuordnen. Sie können mit S3 Access Grants die direkte Zuordnung von S3-Präfixen zu Benutzern und Rollen in Amazon-S3-Buckets und -Objekten definieren. Mit dem vereinfachten Zugriffsschema in S3 Access Grants können Sie beiden IAM Prinzipalen und direkt Benutzern oder Gruppen aus einem Unternehmensverzeichnis Lese-, Schreib- oder Leserechte pro S3-Präfix gewähren. Über diese S3-Access-Grants-Funktionen können Anwendungen im Namen des aktuell authentifizierten Benutzers der Anwendung Daten von Amazon S3 anfordern.

[Wenn Sie S3 Access Grants mit der Funktion zur Verbreitung vertrauenswürdiger Identitäten von integrieren](#) AWS IAM Identity Center, können Ihre Anwendungen Anfragen stellen an AWS-Services (einschließlich S3 Access Grants) direkt im Namen eines authentifizierten Unternehmensverzeichnisbenutzers. Ihre Anwendungen müssen den Benutzer nicht mehr zuerst einem IAM Prinzipal zuordnen. Darüber hinaus wird die Überprüfung vereinfacht, welcher Benutzer auf welches S3-Objekt zugegriffen hat, da Endbenutzeridentitäten bis zu Amazon S3 weitergegeben werden. Sie müssen die Beziehung zwischen verschiedenen Benutzern und IAM Sitzungen nicht mehr rekonstruieren. Wenn Sie S3 Access Grants mit IAM Identity Center Trusted Identity Propagation verwenden, gilt jeweils [AWS CloudTrail](#)Das Datenereignis für Amazon S3 enthält einen direkten Verweis auf den Endbenutzer, in dessen Namen auf die Daten zugegriffen wurde.

Weitere Informationen zu S3 Access Grants finden Sie in den folgenden Themen.

Themen

- [Konzepte von S3 Access Grants](#)
- [S3 Access Grants und Unternehmensverzeichnisidentitäten](#)
- [Erste Schritte mit S3 Access Grants](#)
- [Arbeiten mit S3 Access Grants-Instanzen](#)
- [Mit S3 Access Grants-Standorten arbeiten](#)
- [Arbeiten mit Zuschüssen in S3 Access Grants](#)
- [Abrufen von S3-Daten mithilfe von Zugriffsberechtigungen](#)
- [Kontübergreifender Zugriff mit S3 Access Grants](#)

- [Die Verwendung von AWS Tags mit S3 Access Grants](#)
- [Limits für S3 Access Grants](#)
- [S3-Access-Grants-Integrationen](#)

Konzepte von S3 Access Grants

Arbeitsablauf für S3 Access Grants

Der Arbeitsablauf für S3 Access Grants ist:

1. Erstellen Sie eine S3 Access Grants-Instanz. Siehe [Arbeiten mit S3 Access Grants-Instanzen](#).
2. Registrieren Sie in Ihrer S3 Access Grants-Instance Standorte in Ihren Amazon S3 S3-Daten und ordnen Sie diese Standorte zu AWS Identity and Access Management (IAM) Rollen. Siehe [Registrieren eines Speicherorts](#).
3. Erstellen Sie Zuschüsse für Stipendiaten, die den Stipendiaten Zugriff auf Ihre S3-Ressourcen gewähren. Siehe [Arbeiten mit Zuschüssen in S3 Access Grants](#).
4. Der Empfänger fordert temporäre Anmeldeinformationen von S3 Access Grants an. Siehe [Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants](#).
5. Der Empfänger greift mit diesen temporären Anmeldeinformationen auf die S3-Daten zu. Siehe [Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden](#).

Weitere Informationen finden Sie unter [Erste Schritte mit S3 Access Grants](#).

S3-Access-Grants-Instances

Eine S3 Access Grants-Instanz ist ein logischer Container für einzelne Zuschüsse. Wenn Sie eine S3 Access Grants-Instanz erstellen, müssen Sie eine angeben AWS-Region. Jeder AWS-Region in deinem AWS-Konto kann eine S3 Access Grants-Instanz haben. Weitere Informationen finden Sie unter [Arbeiten mit S3 Access Grants-Instanzen](#).

Wenn Sie S3 Access Grants verwenden möchten, um Zugriff auf Benutzer- und Gruppenidentitäten aus Ihrem Unternehmensverzeichnis zu gewähren, müssen Sie Ihre S3 Access Grants-Instanz auch mit einem verknüpfen AWS IAM Identity Center sein. Weitere Informationen finden Sie unter [S3 Access Grants und Unternehmensverzeichnisidentitäten](#).

Eine neu erstellte S3 Access Grants-Instanz ist leer. Sie müssen einen Speicherort in der Instance registrieren. Dabei kann es sich um den S3-Standardpfad (`s3://`), einen Bucket oder ein Präfix innerhalb eines Buckets handeln. Nachdem Sie mindestens einen Standort registriert haben, können Sie Zugriffsberechtigungen erstellen, die Zugriff auf Daten an diesem registrierten Standort gewähren.

Orte

Ein S3-Zugriff gewährt einen Standort, ordnet Buckets oder Präfixe einem zu AWS Identity and Access Management (IAM) Rolle. S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an den Empfänger weiterzugeben, der auf diesen bestimmten Standort zugreift. Sie müssen zuerst mindestens einen Standort in Ihrer S3 Access Grants-Instanz registrieren, bevor Sie eine Zugriffsberechtigung erstellen können.

Wir empfehlen, den Standardstandort (`s3://`) zu registrieren und ihn einer IAM Rolle zuzuordnen. Der Standort am standardmäßigen S3-Pfad (`s3://`) deckt den Zugriff auf all Ihre S3-Buckets in der AWS-Region Ihres Kontos. Wenn Sie eine Zugriffsberechtigung erstellen, können Sie den Umfang der Gewährung auf einen Bereich, ein Präfix oder ein Objekt innerhalb des Standardverzeichnisses einschränken.

Bei komplexeren Anwendungsfällen für die Zugriffsverwaltung müssen Sie möglicherweise mehr als den Standardspeicherort registrieren. Einige Beispiele für solche Anwendungsfälle sind:

- Nehmen wir an, dass *amzn-s3-demo-bucket* ist ein registrierter Standort in Ihrer S3 Access Grants-Instance, dem eine IAM Rolle zugeordnet ist, aber dieser IAM Rolle wird der Zugriff auf ein bestimmtes Präfix innerhalb des Buckets verweigert. In diesem Fall können Sie das Präfix, auf das die IAM Rolle keinen Zugriff hat, als separaten Standort registrieren und diesen Standort einer anderen IAM Rolle mit dem erforderlichen Zugriff zuordnen.
- Angenommen, Sie möchten Zuweisungen einrichten, die den Zugriff nur auf Benutzer innerhalb eines Virtual Private Cloud (VPC) -Endpunkts beschränken. In diesem Fall können Sie einen Standort für einen Bucket registrieren, in dem die IAM Rolle den Zugriff auf den VPC Endpunkt einschränkt. Wenn später ein Empfänger S3 Access Grants um Anmeldeinformationen bittet, übernimmt S3 Access Grants die IAM Rolle des Standorts, die temporären Anmeldeinformationen zu verkaufen. Diese Anmeldeinformationen verweigern den Zugriff auf den jeweiligen Bucket, sofern sich der Anrufer nicht innerhalb des Endpunkts befindet. VPC Diese Ablehnungsberechtigung wird zusätzlich zu der regulären READ oder in der Gewährung angegebenen READWRITE Berechtigung angewendet. WRITE

Wenn Ihr Anwendungsfall erfordert, dass Sie mehrere Standorte in Ihrer S3 Access Grants-Instanz registrieren, können Sie einen der folgenden Standorte registrieren:

- Der Standard-S3-Standort (`s3://`)
- Ein Bucket (zum Beispiel `amzn-s3-demo-bucket`) oder mehrere Buckets
- Ein Bucket und ein Präfix (zum Beispiel `amzn-s3-demo-bucket/prefix*`) oder mehrere Präfixe

Die maximale Anzahl von Standorten, die Sie in Ihrer S3 Access Grants-Instanz registrieren können, finden Sie unter [Limits für S3 Access Grants](#). Weitere Informationen zur Registrierung eines S3 Access Grants-Standorts finden Sie unter [Registrieren eines Speicherorts](#).

Nachdem Sie den ersten Standort in Ihrer S3 Access Grants-Instanz registriert haben, enthält Ihre Instance immer noch keine individuellen Zugriffsberechtigungen. Es wurde also noch kein Zugriff auf Ihre S3-Daten gewährt. Sie können jetzt Zugriffsberechtigungen erstellen, um Zugriff zu gewähren. Weitere Informationen zum Erstellen von Zuschüssen finden Sie unter [Arbeiten mit Zuschüssen in S3 Access Grants](#).

Gewährungen

Eine individuelle Gewährung in einer S3 Access Grants-Instanz ermöglicht es einer bestimmten Identität — einem IAM Prinzipal, einem Benutzer oder einer Gruppe in einem Unternehmensverzeichnis —, Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist.

Wenn Sie einen Grant erstellen, müssen Sie nicht den Zugriff auf den gesamten registrierten Standort gewähren. Sie können den Zugriffsumfang des Zuschusses innerhalb eines Standorts einschränken. Wenn es sich bei dem registrierten Speicherort um den Standard-S3-Pfad (`s3://`) handelt, müssen Sie den Geltungsbereich des Grants auf einen Bucket, ein Präfix innerhalb eines Buckets oder ein bestimmtes Objekt einschränken. Wenn es sich bei dem registrierten Ort des Grants um einen Bucket oder ein Präfix handelt, können Sie Zugriff auf den gesamten Bucket oder das Präfix gewähren oder optional den Geltungsbereich des Grants auf ein Präfix, ein Unterpräfix oder ein Objekt einschränken.

Im Grant legen Sie auch die Zugriffsebene des Grants auf `READ`, `WRITE`, oder `READWRITE` fest. Angenommen, Sie verfügen über einen Zuschuss, der der Unternehmensverzeichnisgruppe `01234567-89ab-cdef-0123-456789abcdef` `READ` Zugriff auf den Bucket gewährt `s3://amzn-s3-demo-bucket/projects/items/*`. Benutzer in dieser Gruppe können auf jedes Objekt `READ` zugreifen, das einen Objektschlüsselnamen hat, der mit dem Präfix `projects/items/` im Bucket namens beginnt `amzn-s3-demo-bucket`.

Die maximale Anzahl von Grants, die Sie in Ihrer S3 Access Grants-Instanz erstellen können, finden Sie unter [Limits für S3 Access Grants](#). Weitere Informationen zum Erstellen von Zuschüssen finden Sie unter [Erstellen von Gewährungen](#).

Temporäre S3-Access-Grants-Anmeldeinformationen

Nachdem Sie einen Grant erstellt haben, kann eine autorisierte Anwendung, die die in der Grant angegebene Identität verwendet, just-in-time Zugangsdaten anfordern. Zu diesem Zweck ruft die Anwendung den [GetDataAccess](#) API S3-Vorgang auf. Empfänger können diesen API Vorgang verwenden, um Zugriff auf die S3-Daten zu beantragen, die Sie mit ihnen geteilt haben.

Die S3-Access-Grants-Instance evaluiert die [GetDataAccess](#)-Anforderung anhand der verfügbaren Gewährungen. Wenn es einen passenden Zuschuss für den Antragsteller gibt, übernimmt S3 Access Grants die IAM Rolle, die dem registrierten Standort des entsprechenden Zuschusses zugeordnet ist. S3 Access Grants schränkt die Berechtigungen der temporären Anmeldeinformationen so ein, dass sie nur auf den S3-Bucket, das Präfix oder das Objekt zugreifen, die durch den Geltungsbereich der Grant angegeben sind.

Die Ablaufzeit der temporären Zugangsdaten ist standardmäßig auf 1 Stunde eingestellt, Sie können sie jedoch auf einen beliebigen Wert zwischen 15 Minuten und 12 Stunden festlegen. Die maximale Sitzungsdauer finden Sie in der [AssumeRole](#) API Referenz.

Funktionsweise

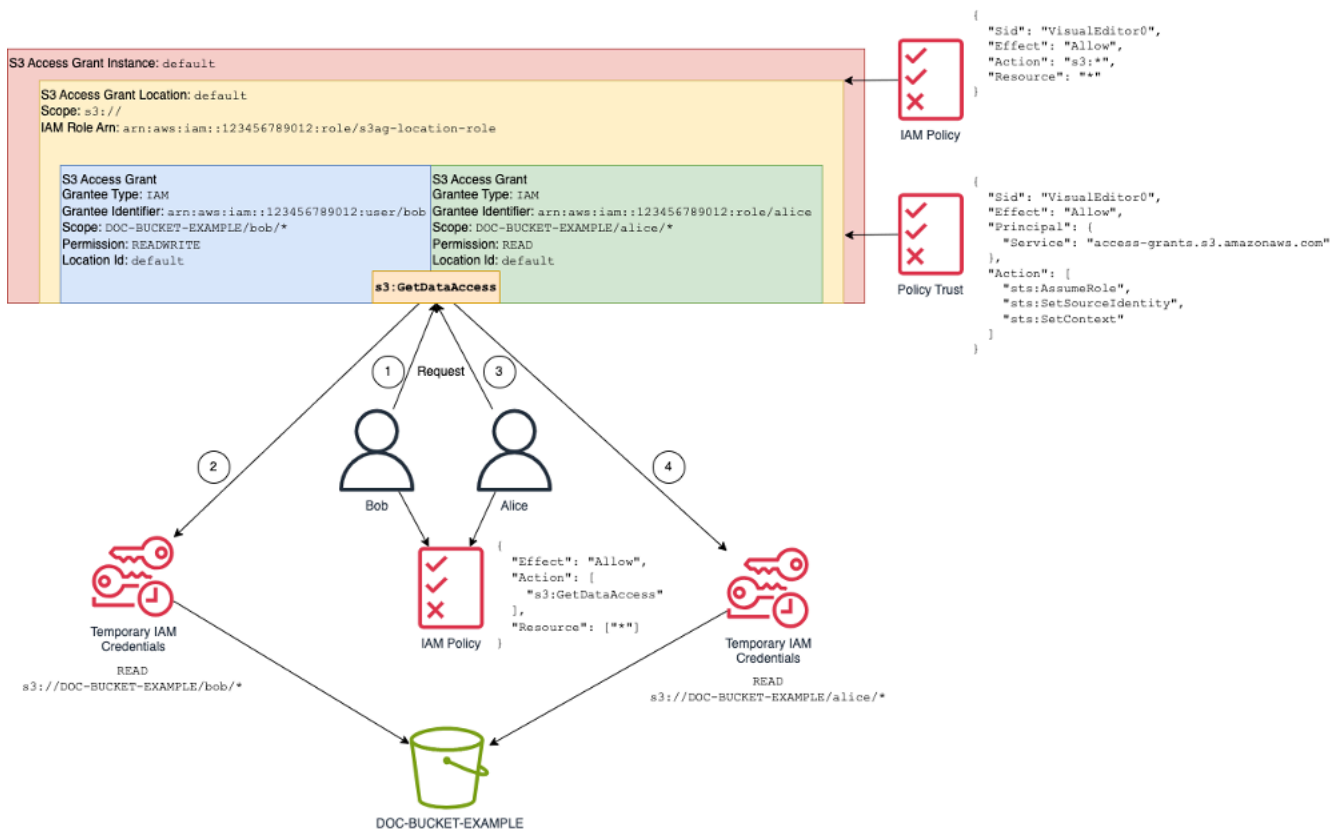
In der folgenden Abbildung `s3://` ist ein Amazon S3 S3-Standardstandort mit dem Geltungsbereich für die IAM Rolle registrierts3ag-location-role. Diese IAM Rolle ist berechtigt, Amazon S3 S3-Aktionen innerhalb des Kontos durchzuführen, wenn ihre Anmeldeinformationen über S3 Access Grants abgerufen wurden.

An diesem Standort werden zwei individuelle Zugriffsberechtigungen für zwei IAM Benutzer erstellt. Dem IAM Benutzer Bob werden READ sowohl der Zugriff als auch der WRITE Zugriff auf das bob/ Präfix im DOC-BUCKET-EXAMPLE Bucket gewährt. Einer anderen IAM Rolle, Alice, wird nur READ Zugriff auf das alice/ Präfix im DOC-BUCKET-EXAMPLE Bucket gewährt. Es wurde eine Gewährung (blau) definiert, damit Bob auf das Präfix bob/ im Bucket DOC-BUCKET-EXAMPLE zugreifen kann. Es wurde eine Gewährung (grün) definiert, damit Alice auf das Präfix alice/ im Bucket DOC-BUCKET-EXAMPLE zugreifen kann.

Wenn Bob READ Daten benötigt, ruft die IAM Rolle, die dem Standort zugeordnet ist, an dem sich sein Grant befindet, den [GetDataAccess](#) API Vorgang S3 Access Grants auf. Wenn Bob

versucht, ein S3-Präfix READ oder Objekt zu verwenden: `s3://DOC-BUCKET-EXAMPLE/bob/*`, mit dem oder dem das Objekt beginnt, gibt die `GetDataAccess` Anfrage einen Satz temporärer IAM Sitzungsanmeldedaten mit der entsprechenden Genehmigung zurück: `s3://DOC-BUCKET-EXAMPLE/bob/*`. Ähnlich kann Bob WRITE-Zugriff für alle S3-Präfixe oder Objekte erhalten, die mit `s3://DOC-BUCKET-EXAMPLE/bob/*` beginnen, da die Gewährung dies ebenfalls zulässt.

Und ähnlich kann Alice READ-Zugriff für alles erhalten, das mit `s3://DOC-BUCKET-EXAMPLE/alice/` beginnt. Wenn sie jedoch versucht, WRITE-Zugriff auf Buckets, Präfixe oder Objekte in `s3://` zu erhalten, wird ihr die Fehlermeldung „Access Denied“ (403 Forbidden) angezeigt, da es keine Gewährung gibt, die ihr einen WRITE-Zugriff auf Daten gewährt. Wenn Alice Zugriff (READ oder WRITE) auf Daten außerhalb von `s3://DOC-BUCKET-EXAMPLE/alice/` anfordert, wird ihr ebenfalls die Fehlermeldung „Access Denied“ angezeigt.



Dieses Muster kann auf eine große Zahl von Benutzern und Buckets skaliert werden und vereinfacht die Verwaltung dieser Berechtigungen. Anstatt jedes Mal, wenn Sie individuelle Benutzer-Präfix-Zugriffsbeziehungen hinzufügen oder entfernen möchten, potenziell umfangreiche S3-Bucket-Richtlinien zu bearbeiten, können Sie einzelne, diskrete Gewährungen hinzufügen und entfernen.

S3 Access Grants und Unternehmensverzeichnisidentitäten

Sie können Amazon S3 Access Grants verwenden, um Zugriff zu gewähren AWS Identity and Access Management (IAM) Principals (Benutzer oder Rollen), beide im selben AWS-Konto und in anderen. Häufig handelt es sich jedoch bei der Entität, die auf die Daten zugreift, um Endbenutzer in Ihrem Unternehmensverzeichnis. Anstatt den Hauptbenutzern Zugriff IAM zu gewähren, können Sie S3 Access Grants verwenden, um Ihren Unternehmensbenutzern und Gruppen direkt Zugriff zu gewähren. Mit S3 Access Grants müssen Sie Ihre Unternehmensidentitäten nicht mehr zwischengeschalteten IAM Prinzipalen zuordnen, um über Ihre Unternehmensanwendungen auf Ihre S3-Daten zuzugreifen.

Diese neue Funktionalität — Unterstützung für den Zugriff auf Daten mithilfe von Endbenutzeridentitäten — wird bereitgestellt, indem Sie Ihre S3 Access Grants-Instanz mit einem verknüpfen AWS IAM Identity Center sein. IAM Identity Center unterstützt standardbasierte Identitätsanbieter und ist der Dreh- und Angelpunkt für AWS für alle Dienste oder Funktionen, einschließlich S3 Access Grants, die Endbenutzeridentitäten unterstützen. IAM Identity Center bietet über seine Funktion zur Weitergabe vertrauenswürdiger Identitäten Unterstützung bei der Authentifizierung von Unternehmensidentitäten. Weitere Informationen finden Sie unter [Vertrauenswürdige Identitätsweitergabe über Anwendungen hinweg](#).

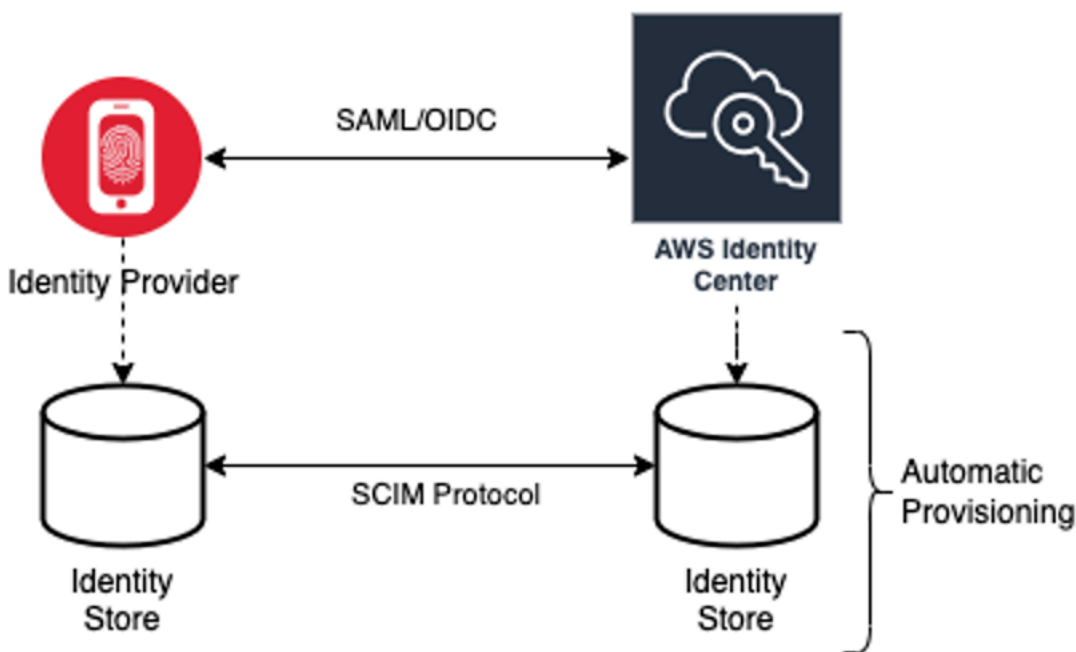
Um mit der Unterstützung von Personalidentitäten in S3 Access Grants zu beginnen, müssen Sie zunächst in IAM Identity Center die Identitätsbereitstellung zwischen Ihrem Corporate Identity Provider und IAM Identity Center konfigurieren. IAM Identity Center unterstützt Anbieter von Unternehmensidentitäten wie Okta, Microsoft Entra ID (früher Azure Active Directory) oder einen anderen externen Identitätsanbieter (IdP), der das System for Cross-Domain Identity Management (SCIM) -Protokoll unterstützt. Wenn Sie IAM Identity Center mit Ihrem IdP verbinden und die automatische Bereitstellung aktivieren, werden die Benutzer und Gruppen Ihres IdP mit dem Identitätsspeicher in IAM Identity Center synchronisiert. Nach diesem Schritt verfügt IAM Identity Center über eine eigene Ansicht Ihrer Benutzer und Gruppen, sodass Sie mithilfe anderer Optionen auf sie verweisen können AWS-Services und Funktionen wie S3 Access Grants. Weitere Informationen zur Konfiguration der automatischen Bereitstellung von IAM Identity Center finden Sie unter [Automatische Bereitstellung](#) im AWS IAM Identity Center Benutzerleitfaden.

IAM Identity Center ist integriert in AWS Organizations sodass Sie Berechtigungen für mehrere zentral verwalten können AWS-Konten ohne jedes Ihrer Konten manuell zu konfigurieren. In einer typischen Organisation konfiguriert Ihr Identitätsadministrator eine IAM Identity Center-Instanz für die gesamte Organisation als zentralen Punkt der Identitätssynchronisierung. Diese IAM Identity Center-Instanz wird normalerweise in einer dedizierten Instanz ausgeführt AWS-Konto in Ihrer Organisation.

In dieser allgemeinen Konfiguration können Sie von jedem beliebigen Ort aus auf Benutzer- und Gruppenidentitäten in S3 Access Grants verweisen AWS-Konto in der Organisation.

Wenn jedoch dein AWS Organizations Der Administrator hat noch keine zentrale IAM Identity Center-Instanz konfiguriert, Sie können eine lokale Instanz im selben Konto wie Ihre S3 Access Grants-Instanz erstellen. Eine solche Konfiguration ist eher für proof-of-concept Anwendungsfälle in der lokalen Entwicklung üblich. In allen Fällen muss sich die IAM Identity Center-Instanz in derselben befinden AWS-Region wie die S3 Access Grants-Instanz, der sie zugeordnet wird.

In der folgenden Abbildung einer IAM Identity Center-Konfiguration mit einem externen IdP ist der IdP so konfiguriert, dass er den Identitätsspeicher vom IdP mit SCIM dem Identitätsspeicher in IAM Identity Center synchronisiert.



Gehen Sie wie folgt vor, um Ihre Unternehmensverzeichnisidentitäten mit S3 Access Grants zu verwenden:

- Richten Sie die [automatische Bereitstellung](#) in IAM Identity Center ein, um Benutzer- und Gruppeninformationen von Ihrem IdP mit IAM Identity Center zu synchronisieren.
- Konfigurieren Sie Ihre externe Identitätsquelle innerhalb von IAM Identity Center als vertrauenswürdigen Token-Aussteller. Weitere Informationen finden Sie unter [Anwendungsübergreifende Verbreitung vertrauenswürdiger Identitäten](#) im AWS IAM Identity Center Benutzerleitfaden.
- Verknüpfen Sie Ihre S3 Access Grants-Instanz mit Ihrer IAM Identity Center-Instanz. Sie können dies während der [Erstellung Ihrer S3-Access-Grants-Instance](#) durchführen. Wenn Sie Ihre S3-

Access-Grants-Instance bereits erstellt haben, finden Sie weitere Informationen unter [Ordnen Sie Ihre IAM Identity Center-Instanz zu oder trennen Sie sie.](#)

Zugriff auf S3-Daten über Verzeichnisidentitäten

Angenommen, Sie haben Unternehmensverzeichnisbenutzer, die über eine Unternehmensanwendung auf Ihre S3-Daten zugreifen müssen, z. B. eine Anwendung zur Dokumentenanzeige, die in Ihren externen IdP integriert ist (z. B. Okta), um Benutzer zu authentifizieren. Die Authentifizierung des Benutzers in diesen Anwendungen erfolgt in der Regel über Weiterleitungen im Webbrowser des Benutzers. Da es sich bei den Benutzern im Verzeichnis nicht um IAM Principals handelt, benötigt Ihre Anwendung IAM Anmeldeinformationen, mit denen sie den GetDataAccess API Vorgang S3 Access Grants aufrufen kann, [um im Namen der Benutzer Anmeldeinformationen für S3-Daten](#) abzurufen. Im Gegensatz zu IAM Benutzern und Rollen, die die Anmeldeinformationen selbst abrufen, muss Ihre Anwendung die Möglichkeit haben, einen Verzeichnisbenutzer darzustellen, der keiner IAM Rolle zugeordnet ist, sodass der Benutzer Datenzugriff über S3 Access Grants erhalten kann.

Dieser Übergang vom authentifizierten Verzeichnisbenutzer zu einem IAM Anrufer, der im Namen des Verzeichnisbenutzers Anfragen an S3 Access Grants stellen kann, erfolgt durch die Anwendung über die Funktion Trusted Token Issuer von Identity Center. IAM Die Anwendung hat nach der Authentifizierung des Verzeichnisbenutzers ein Identitätstoken vom IdP (z. B. Okta), das den Verzeichnisbenutzer darstellt gemäß Okta. Die Konfiguration des vertrauenswürdigen Token-Ausstellers in IAM Identity Center ermöglicht es der Anwendung, diese auszutauschen Okta Token (das Okta Der Mandant ist als „vertrauenswürdiger Aussteller“ ()) für ein anderes Identitätstoken von IAM Identity Center konfiguriert, das den Verzeichnisbenutzer innerhalb sicher repräsentiert AWS-Services. Die Datenanwendung übernimmt dann eine IAM Rolle und stellt das Token des Verzeichnisbenutzers aus IAM Identity Center als zusätzlichen Kontext bereit. Die Anwendung kann die resultierende IAM Sitzung verwenden, um S3 Access Grants aufzurufen. Das Token stellt sowohl die Identität der Anwendung (den IAM Prinzipal selbst) als auch die Identität des Verzeichnisbenutzers dar.

Der Hauptschritt dieses Wechsels ist der Token-Austausch. Die Anwendung führt diesen Token-Austausch durch, indem sie den CreateTokenWithIAM API Vorgang in IAM Identity Center aufruft. Natürlich ist auch das ein AWS APIruft an und verlangt von einem IAM Schulleiter, dass er es unterschreibt. Der IAM Principal, der diese Anfrage stellt, ist in der Regel eine IAM Rolle, die der Anwendung zugeordnet ist. Wenn die Anwendung beispielsweise auf Amazon ausgeführt wird EC2, wird die CreateTokenWithIAM Anfrage normalerweise von der IAM Rolle ausgeführt, die der EC2 Instance zugeordnet ist, auf der die Anwendung ausgeführt wird. Das Ergebnis eines erfolgreichen

`CreateTokenWithIAM` Aufrufs ist ein neues Identitätstoken, das innerhalb erkannt wird AWS-Services.

Der nächste Schritt, bevor die Anwendung im Namen des Verzeichnisbenutzers aufrufen `GetDataAccess` kann, besteht darin, dass die Anwendung eine IAM Sitzung aufruft, die die Identität des Verzeichnisbenutzers enthält. Die Anwendung tut dies mit einem AWS Security Token Service (AWS STS) `AssumeRole` Anfrage, die auch das IAM Identity Center-Token für den Verzeichnisbenutzer als zusätzlichen Identitätskontext enthält. Dieser zusätzliche Kontext ermöglicht es IAM Identity Center, die Identität des Verzeichnisbenutzers an den nächsten Schritt weiterzugeben. Die IAM Rolle, die die Anwendung annimmt, ist die Rolle, für die IAM Berechtigungen zum Aufrufen des `GetDataAccess` Vorgangs erforderlich sind.

Nachdem die Anwendung die IAM Rolle des Identitätsträgers mit dem IAM Identity Center-Token für den Verzeichnisbenutzer als zusätzlichen Kontext übernommen hat, verfügt sie nun über alles, was sie benötigt, um im Namen des authentifizierten Verzeichnisbenutzers eine signierte Anfrage zu `GetDataAccess` stellen.

Die Token-Weitergabe basiert auf den folgenden Schritten:

Erstellen Sie eine IAM Identity Center-Anwendung

Erstellen Sie zunächst eine neue Anwendung in IAM Identity Center. Diese Anwendung verwendet eine Vorlage, anhand derer IAM Identity Center ermitteln kann, welche Art von Anwendungseinstellungen Sie verwenden können. Für den Befehl zum Erstellen der Anwendung müssen Sie den Amazon Resource Name (ARN) der IAM Identity Center-Instance, einen Anwendungsnamen und den Anwendungsanbieter angeben. Der Anwendungsanbieter ist der SAML oder der OAuth Anwendungsanbieter, den die Anwendung verwendet, um IAM Identity Center aufzurufen.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch eigenen Daten:

```
aws sso-admin create-application \  
  --instance-arn "arn:aws:sso:::instance/ssoins-ssoins-1234567890abcdef" \  
  --application-provider-arn "arn:aws:sso::aws:applicationProvider/custom" \  
  --name MyDataApplication
```

Antwort:

```
{
```

```
"ApplicationArn": "arn:aws:sso::123456789012:application/ssoins-  
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"  
}
```

Erstellen eines vertrauenswürdigen Token-Ausstellers

Jetzt, wo Sie Ihre IAM Identity Center-Anwendung haben, besteht der nächste Schritt darin, einen vertrauenswürdigen Token-Aussteller zu konfigurieren, der verwendet wird, um Ihre IdToken Werte von Ihrem IdP gegen IAM Identity Center-Token auszutauschen. In diesem Schritt müssen Sie die folgenden Elemente bereitstellen:

- Der Emittent des Identitätsanbieters URL
- Name des vertrauenswürdigen Token-Ausstellers
- Pfad zum Anforderungsattribut
- Pfad zum Identitätsspeicherattribut
- Die Abrufoption JSON Web Key Set (JWKS)

Der Pfad zum Anforderungsattribut ist das Identitätsanbieterattribut, das für die Zuordnung zum Identitätsspeicherattribut verwendet wird. Normalerweise ist der Pfad zum Anforderungsattribut die E-Mail-Adresse des Benutzers. Sie können jedoch auch andere Attribute für die Zuordnung verwenden.

Erstellen Sie eine Datei mit dem Namen `oidc-configuration.json` und den folgenden Informationen. Wenn Sie diese Datei verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Daten.

```
{  
  "OidcJwtConfiguration":  
    {  
      "IssuerUrl": "https://login.microsoftonline.com/a1b2c3d4-abcd-1234-b7d5-  
b154440ac123/v2.0",  
      "ClaimAttributePath": "preferred_username",  
      "IdentityStoreAttributePath": "userName",  
      "JwksRetrievalOption": "OPEN_ID_DISCOVERY"  
    }  
}
```

Führen Sie den folgenden Befehl aus, um den vertrauenswürdigen Token-Aussteller zu erstellen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws sso-admin create-trusted-token-issuer \  
  --instance-arn "arn:aws:sso:::instance/ssoins-1234567890abcdef" \  
  --name MyEntraIDTrustedIssuer \  
  --trusted-token-issuer-type OIDC_JWT \  
  --trusted-token-issuer-configuration file://./oidc-configuration.json
```

Antwort

```
{  
  "TrustedTokenIssuerArn": "arn:aws:sso:::123456789012:trustedTokenIssuer/  
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234"  
}
```

Connect die IAM Identity Center-Anwendung mit dem vertrauenswürdigen Token-Aussteller

Für den vertrauenswürdigen Token-Aussteller sind einige weitere Konfigurationseinstellungen erforderlich, um zu funktionieren. Legen Sie die Zielgruppe fest, der der vertrauenswürdige Token-Aussteller vertrauen soll. Die Zielgruppe ist der Wert im IdToken, der durch den Schlüssel identifiziert wird und in den Einstellungen des Identitätsanbieters zu finden ist. Beispielsweise:

```
1234973b-abcd-1234-abcd-345c5a9c1234
```

Erstellen Sie eine Datei mit dem Namen `grant.json` mit dem folgenden Inhalt. Um diese Datei zu verwenden, ändern Sie die Zielgruppe so, dass sie Ihren Identitätsanbieter-Einstellungen entspricht, und geben Sie den vertrauenswürdigen Token-Aussteller an ARN, der mit dem vorherigen Befehl zurückgegeben wurde.

```
{  
  "JwtBearer":  
    {  
      "AuthorizedTokenIssuers":  
        [  
          {  
            "TrustedTokenIssuerArn": "arn:aws:sso:::123456789012:trustedTokenIssuer/  
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234",  
            "AuthorizedAudiences":  
              [  
                "1234973b-abcd-1234-abcd-345c5a9c1234"  
              ]  
          }  
        ]  
    }  
}
```



```
]
}
}
```

Führen Sie den folgenden Beispielbefehl aus. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws sso-admin put-application-grant \  
  --application-arn "arn:aws:sso::123456789012:application/ssoins-  
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \  
  --grant-type "urn:ietf:params:oauth:grant-type:jwt-bearer" \  
  --grant file://./grant.json \  

```

Mit diesem Befehl werden die Konfigurationseinstellungen des vertrauenswürdigen Token-Ausstellers so festgelegt, dass er der Zielgruppe in der Datei `grant.json` vertraut und diese Zielgruppe mit der Anwendung verknüpft, die im ersten Schritt für den Austausch von Token des Typs `jwt-bearer` erstellt wurde. Die Zeichenfolge `urn:ietf:params:oauth:grant-type:jwt-bearer` ist keine willkürliche Zeichenfolge. Es ist ein registrierter Namespace in OAuth JSON Web Token (JWT) - Assertion-Profilen. [Weitere Informationen zu diesem Namespace finden Sie in 7523. RFC](#)

Verwenden Sie als Nächstes den folgenden Befehl, um festzulegen, welche Bereiche der vertrauenswürdige Token-Aussteller beim Austausch von IdToken-Werten Ihres Identitätsanbieters einbezieht. Für S3 Access Grants ist der Wert für den Parameter `--scope` `s3:access_grants:read_write`.

```
aws sso-admin put-application-access-scope \  
  --application-arn "arn:aws:sso::111122223333:application/ssoins-  
ssoins-111122223333abcdef/apl-abcd1234a1b2c3d" \  
  --scope "s3:access_grants:read_write"
```

Der letzte Schritt besteht darin, der IAM Identity Center-Anwendung eine Ressourcenrichtlinie anzuhängen. Diese Richtlinie ermöglicht es Ihrer IAM Anwendungsrolle, Anfragen an den API Vorgang zu stellen `sso-oauth:CreateTokenWithIAM` und die IdToken Werte von IAM Identity Center zu empfangen.

Erstellen Sie eine Datei mit dem Namen `authentication-method.json` mit folgendem Inhalt. Ersetzen Sie *123456789012* durch Ihre Konto-ID.

```
{
```

```
"Iam":
  {
    "ActorPolicy":
      {
        "Version": "2012-10-17",
        "Statement":
          [
            {
              "Effect": "Allow",
              "Principal":
                {
                  "AWS": "arn:aws:iam::123456789012:role/webapp"
                },
              "Action": "sso-oauth:CreateTokenWithIAM",
              "Resource": "*"
            }
          ]
      }
  }
}
```

Führen Sie den folgenden Befehl aus, um die Richtlinie an die IAM Identity Center-Anwendung anzuhängen:

```
aws sso-admin put-application-authentication-method \
  --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
  --authentication-method-type IAM \
  --authentication-method file://./authentication-method.json
```

Damit sind die Konfigurationseinstellungen für die Verwendung von S3 Access Grants mit Verzeichnisbenutzern über eine Webanwendung abgeschlossen. Sie können dieses Setup direkt in der Anwendung testen oder den CreateTokenWithIAM API Vorgang aufrufen, indem Sie den folgenden Befehl von einer in der IAM Identity Center-Anwendungsrichtlinie zugelassenen IAM Rolle aus verwenden:

```
aws sso-oidc create-token-with-iam \
  --client-id "arn:aws:sso::123456789012:application/ssoins-ssoins-1234567890abcdef/
apl-abcd1234a1b2c3d" \
  --grant-type urn:ietf:params:oauth:grant-type:jwt-bearer \
  --assertion IdToken
```

Die Antwort wird sieht ungefähr wie folgt aus:

```
{
  "accessToken": "<suppressed long string to reduce space>",
  "tokenType": "Bearer",
  "expiresIn": 3600,
  "refreshToken": "<suppressed long string to reduce space>",
  "idToken": "<suppressed long string to reduce space>",
  "issuedTokenType": "urn:ietf:params:oauth:token-type:refresh_token",
  "scope": [
    "sts:identity_context",
    "s3:access_grants:read_write",
    "openid",
    "aws"
  ]
}
```

Wenn Sie den mit Base64 codierten IdToken Wert dekodieren, können Sie die Schlüssel-Wert-Paare im Format sehen. JSON Der Schlüssel `sts:identity_context` enthält den Wert, den Ihre Anwendung in der `sts:AssumeRole`-Anforderung senden muss, um die Identitätsinformationen des Verzeichnisbenutzers einzuschließen. Dies ist ein Beispiel für das dekodierte IdToken:

```
{
  "aws:identity_store_id": "d-996773e796",
  "sts:identity_context": "AQoJb3JpZ2luX2VjE0Tt1;<SUPRESSED>",
  "sub": "83d43802-00b1-7054-db02-f1d683aacba5",
  "aws:instance_account": "123456789012",
  "iss": "https://identitycenter.amazonaws.com/ssoins-1234567890abcdef",
  "sts:audit_context": "AQoJb3JpZ2luX2VjE0T<SUPRESSED>==",
  "aws:identity_store_arn": "arn:aws:identitystore::232642235904:identitystore/d-996773e796",
  "aud": "abcd12344U0gi7n4Yyp0-WV1LWN1bnRyYWwtMQ",
  "aws:instance_arn": "arn:aws:sso:::instance/ssoins-6987d7fb04cf7a51",
  "aws:credential_id": "EXAMPLEHI5glPh40y9TpApJn8...",
  "act": {
    "sub": "arn:aws:sso::232642235904:trustedTokenIssuer/ssoins-6987d7fb04cf7a51/43b4a822-1020-7053-3631-cb2d3e28d10e"
  },
  "auth_time": "2023-11-01T20:24:28Z",
  "exp": 1698873868,
  "iat": 1698870268
}
```

Sie können den Wert aus `sts:identity_context` abrufen und diese Informationen in einem `sts:AssumeRole`-Aufruf weitergeben. Das Folgende ist ein CLI Beispiel für die Syntax. Bei der Rolle, die übernommen werden soll, handelt es sich um eine temporäre Rolle mit Berechtigungen für den Aufruf von `s3:GetDataAccess`.

```
aws sts assume-role \  
  --role-arn "arn:aws:iam::123456789012:role/temp-role" \  
  --role-session-name "TempDirectoryUserRole" \  
  --provided-contexts ProviderArn="arn:aws:iam::aws:contextProvider/  
IdentityCenter",ContextAssertion="value from sts:identity_context"
```

Sie können nun die bei diesem Aufruf erhaltenen Anmeldeinformationen verwenden, um den `s3:GetDataAccess` API Vorgang aufzurufen und die endgültigen Anmeldeinformationen für den Zugriff auf Ihre S3-Ressourcen zu erhalten.

Erste Schritte mit S3 Access Grants

Amazon S3 Access Grants ist eine Amazon-S3-Funktion, die eine skalierbare Lösung für die Zugriffssteuerung für Ihre S3-Daten bereitstellt. S3 Access Grants ist ein Anbieter von S3-Anmeldeinformationen. Das bedeutet, dass Sie Ihre Liste von Gewähungen und deren Zugriffsebenen bei S3 Access Grants registrieren. Wenn Benutzer oder Clients anschließend Zugriff auf Ihre S3-Daten benötigen, fordern sie zunächst Anmeldeinformationen von S3 Access Grants an. Wenn es eine entsprechende Gewährung gibt, die den Zugriff autorisiert, stellt S3 Access Grants temporäre Zugangsdaten mit geringsten Berechtigungen bereit. Die Benutzer oder Clients können dann mit den von S3 Access Grants bereitgestellten Anmeldeinformationen auf Ihre S3-Daten zuzugreifen. Wenn Ihre S3-Datenanforderungen eine komplexe oder umfangreiche Berechtigungskonfiguration erfordern, können Sie mit S3 Access Grants die S3-Datenberechtigungen für Benutzer, Gruppen, Rollen und Anwendungen skalieren.

In den meisten Anwendungsfällen können Sie die Zugriffskontrolle für Ihre S3-Daten verwalten, indem Sie AWS Identity and Access Management (IAM) mit Bucket-Richtlinien oder IAM identitätsbasierten Richtlinien.

Wenn es jedoch komplexe Anforderungen an die S3-Zugriffssteuerung gibt, z. B. die folgenden, könnten Sie erheblich von S3 Access Grants profitieren:

- Sie erreichen das Größenlimit von 20 KB für die Bucket-Richtlinie.
- Sie gewähren menschliche Identitäten, zum Beispiel Microsoft Entra ID (früher Azure Active Directory), Okta, oder Ping Benutzer und Gruppen, Zugriff auf S3-Daten für Analysen und Big Data.

- Sie müssen kontoübergreifenden Zugriff gewähren, ohne die IAM Richtlinien häufig aktualisieren zu müssen.
- Ihre Daten sind unstrukturiert und befinden sich eher auf Objektebene, statt ein strukturiertes Zeilen- und Spaltenformat aufzuweisen.

Der Workflow in S3 Access Grants ist wie folgt:

Schritte	Beschreibung
1	<p>Erstellen einer S3-Access-Grants-Instance</p> <p>Initiieren Sie zunächst eine S3-Access-Grants-Instance, die Ihre individuellen Zugriffsberechtigungen enthält.</p>
2	<p>Registrieren eines Speicherorts</p> <p>Zweitens registrieren Sie einen S3-Datenstandort (z. B. den Standardspeicherorts3 : //) und geben Sie dann eine Standardrolle an, die S3 Access Grants IAM bei der Bereitstellung des Zugriffs auf den S3-Datenstandort übernimmt. Sie können bestimmten Buckets oder Präfixen auch benutzerdefinierte Speicherorte hinzufügen und diese benutzerdefinierten IAM Rollen zuordnen.</p>
3	<p>Erstellen von Gewährungen</p> <p>Erstellen Sie individuelle Berechtigungsgewährungen. Geben Sie in diesen Berechtigungsgewährungen den registrierten S3-Speicherort, den Umfang des Datenzugriffs innerhalb des Speicherorts, die Identität des Gewährungsempfängers und die Zugriffsebene (READ, WRITE oder READWRITE) an.</p>
4	<p>Anfordern von Zugriff auf S3-Daten</p> <p>Wenn Benutzer, Anwendungen und AWS-Services möchten auf S3-Daten zugreifen, stellen sie zunächst eine Zugriffsanfrage. S3 Access Grants legt fest, ob die Anforderung autorisiert werden soll. Wenn es eine entsprechende Genehmigung gibt,</p>

Schritte	Beschreibung
	die den Zugriff autorisiert, verwendet S3 Access Grants die IAM Rolle des registrierten Standorts, die mit dieser Genehmigung verknüpft ist, um temporäre Anmeldeinformationen an den Antragsteller zurückzugeben.
5	Zugriff auf S3-Daten Anwendungen verwenden für den Zugriff auf S3-Daten die temporären Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden.

Arbeiten mit S3 Access Grants-Instanzen

Um mit der Verwendung von Amazon S3 Access Grants zu beginnen, erstellen Sie zunächst eine S3-Access-Grants-Instance. Sie können jeweils nur eine S3 Access Grants-Instanz erstellen AWS-Region pro Konto. Die S3-Access-Grants-Instance dient als Container für Ihre S3-Access-Grants-Ressourcen, zu denen registrierte Speicherorte und Gewährungen gehören.

Mit S3 Access Grants können Sie Genehmigungen für Ihre S3-Daten erstellen für AWS Identity and Access Management (IAM) Benutzer und Rollen. Wenn Sie [Ihr Corporate-Identity-Verzeichnis hinzugefügt](#) haben AWS IAM Identity Center, können Sie diese IAM Identity Center-Instanz Ihres Unternehmensverzeichnisses mit Ihrer S3 Access Grants-Instanz verknüpfen. Anschließend können Sie Zugriffsgewährungen für die Benutzer und Gruppen Ihres Unternehmens erstellen. Wenn Sie Ihr Unternehmensverzeichnis noch nicht zu IAM Identity Center hinzugefügt haben, können Sie Ihre S3 Access Grants-Instanz später mit einer IAM Identity Center-Instanz verknüpfen.

Themen

- [Erstellen einer S3-Access-Grants-Instance](#)
- [Rufen Sie die Details einer S3 Access Grants-Instance ab](#)
- [Listen Sie Ihre S3 Access Grants-Instanzen auf](#)
- [Ordnen Sie Ihre IAM Identity Center-Instanz zu oder trennen Sie sie](#)
- [Löschen einer S3-Access-Grants-Instance](#)

Erstellen einer S3-Access-Grants-Instance

Um mit der Verwendung von Amazon S3 Access Grants zu beginnen, erstellen Sie zunächst eine S3-Access-Grants-Instance. Sie können jeweils nur eine S3 Access Grants-Instanz erstellen AWS-Region pro Konto. Die S3-Access-Grants-Instance dient als Container für Ihre S3-Access-Grants-Ressourcen, zu denen registrierte Speicherorte und Gewährungen gehören.

Mit S3 Access Grants können Sie Genehmigungen für Ihre S3-Daten erstellen für AWS Identity and Access Management (IAM) Benutzer und Rollen. Wenn Sie [Ihr Corporate-Identity-Verzeichnis hinzugefügt](#) haben AWS IAM Identity Center, können Sie diese IAM Identity Center-Instanz Ihres Unternehmensverzeichnisses mit Ihrer S3 Access Grants-Instanz verknüpfen. Anschließend können Sie Zugriffsgewährungen für die Benutzer und Gruppen Ihres Unternehmens erstellen. Wenn Sie Ihr Unternehmensverzeichnis noch nicht zu IAM Identity Center hinzugefügt haben, können Sie Ihre S3 Access Grants-Instanz später mit einer IAM Identity Center-Instanz verknüpfen.

Sie können eine S3 Access Grants-Instance mithilfe der Amazon S3 S3-Konsole erstellen, AWS Command Line Interface (AWS CLI), Amazon S3 REST API und AWS SDKs.

Verwenden der S3-Konsole

Bevor Sie mit S3 Access Grants Zugriff auf Ihre S3-Daten gewähren können, müssen Sie zunächst in derselben Instanz eine S3 Access Grants-Instance erstellen AWS-Region wie Ihre S3-Daten.

Voraussetzungen

Wenn Sie mithilfe von Identitäten aus Ihrem Unternehmensverzeichnis Zugriff auf Ihre S3-Daten gewähren möchten, [fügen Sie Ihr Corporate-Identity-Verzeichnis](#) zu AWS IAM Identity Center. Wenn Sie dazu noch nicht bereit sind, können Sie Ihre S3 Access Grants-Instanz später mit einer IAM Identity Center-Instanz verknüpfen.

So erstellen Sie eine S3-Access-Grants-Instance

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste den Namen der aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
4. Wählen Sie auf der Seite S3 Access Grants die Option S3-Access-Grants-Instance erstellen aus.

- a. Stellen Sie in Schritt 1 des Assistenten zum Einrichten der Access Grants-Instanz sicher, dass Sie die Instanz in der aktuellen Version erstellen möchten AWS-Region. Stellen Sie sicher, dass dies dasselbe ist AWS-Region wo sich Ihre S3-Daten befinden. Sie können pro eine S3 Access Grants-Instanz erstellen AWS-Region pro Konto.
- b. (Optional) Wenn Sie [Ihr Corporate-Identity-Verzeichnis hinzugefügt](#) haben AWS IAM Identity Center, können Sie diese IAM Identity Center-Instanz Ihres Unternehmensverzeichnisses mit Ihrer S3 Access Grants-Instanz verknüpfen.

Wählen Sie dazu IAM Identity Center-Instanz hinzufügen in **region**. Geben Sie dann den Amazon Resource Name (ARN) der IAM Identity Center-Instanz ein.

Wenn Sie Ihr Unternehmensverzeichnis noch nicht zu IAM Identity Center hinzugefügt haben, können Sie Ihre S3 Access Grants-Instance später mit einer IAM Identity Center-Instance verknüpfen.

- c. Um die S3-Access-Grants-Instance zu erstellen, wählen Sie Weiter aus. Informationen zur Registrierung eines Speicherorts finden Sie unter [Schritt 2 – Registrieren eines Speicherorts](#).
5. Wenn Weiter oder S3-Access-Grants-Instance erstellen deaktiviert ist:

Instance kann nicht erstellt werden

- Möglicherweise haben Sie bereits eine S3 Access Grants-Instanz in derselben AWS-Region. Wählen Sie im linken Navigationsbereich Access Grants aus. Scrollen Sie auf der Seite S3 Access Grants nach unten zum Abschnitt S3-Access-Grants-Instance in Ihrem Konto, um festzustellen, ob es bereits eine Instance gibt.
- Möglicherweise verfügen Sie nicht über die Berechtigung `s3:CreateAccessGrantsInstance`, die für die Erstellung einer S3-Access-Grants-Instance erforderlich ist. Nehmen Sie Kontakt mit Ihrem Kontoadministrator auf. Zusätzliche Berechtigungen, die erforderlich sind, wenn Sie Ihrer S3 Access Grants-Instanz eine IAM Identity Center-Instanz zuordnen, finden Sie unter [CreateAccessGrantsInstance](#).

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example Erstellen einer S3-Access-Grants-Instance

```
aws s3control create-access-grants-instance \  
--account-id 111122223333 \  
--region us-east-2
```

Antwort:

```
{  
  "CreatedAt": "2023-05-31T17:54:07.893000+00:00",  
  "AccessGrantsInstanceId": "default",  
  "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default"  
}
```

Mit dem REST API

Sie können Amazon S3 verwenden RESTAPI, um eine S3 Access Grants-Instance zu erstellen. Informationen zur REST API Unterstützung bei der Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [AssociateAccessGrantsIdentityCenter](#)
- [CreateAccessGrantsInstance](#)
- [DeleteAccessGrantsInstance](#)
- [DissociateAccessGrantsIdentityCenter](#)
- [GetAccessGrantsInstance](#)
- [GetAccessGrantsInstanceForPrefix](#)
- [GetAccessGrantsInstanceResourcePolicy](#)
- [ListAccessGrantsInstances](#)
- [PutAccessGrantsInstanceResourcePolicy](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält ein Beispiel für die Erstellung einer S3 Access Grants-Instance mithilfe der AWS SDKs.

Java

In diesem Beispiel wird die S3-Access-Grants-Instance erstellt, die als Container für Ihre individuellen Zugriffsgewährungen dient. Sie können eine S3 Access Grants-Instanz pro AWS-Region in Ihrem Konto erstellen. Die Antwort enthält die Instance-ID default und einen Amazon-Ressourcennamen (ARN), der für Ihre S3 Access Grants-Instance generiert wurde.

Example Erstellen einer S3-Access-Grants-Instance-Anforderung

```
public void createAccessGrantsInstance() {
    CreateAccessGrantsInstanceRequest createRequest =
        CreateAccessGrantsInstanceRequest.builder().accountId("111122223333").build();
    CreateAccessGrantsInstanceResponse createResponse =
        s3Control.createAccessGrantsInstance(createRequest);LOGGER.info("CreateAccessGrantsInstance
" + createResponse);
}
```

Antwort:

```
CreateAccessGrantsInstanceResponse(
    CreatedAt=2023-06-07T01:46:20.507Z,
    AccessGrantsInstanceId=default,
    AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default)
```

Rufen Sie die Details einer S3 Access Grants-Instance ab

Sie können die Details Ihrer Amazon S3 Access Grants-Instance in einer bestimmten AWS-Region abrufen. Sie können die Details Ihrer S3 Access Grants-Instance abrufen, indem Sie die Amazon S3 S3-Konsole verwenden, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die AWS SDKs.

Verwenden der S3-Konsole

Um die Details einer S3 Access Grants-Instance abzurufen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

4. Auf der Seite S3 Access Grants werden Ihre S3-Access-Grants-Instances und alle kontoübergreifenden Instances aufgeführt, die mit Ihrem Konto geteilt wurden. Um die Details einer Instance anzuzeigen, wählen Sie Details anzeigen aus.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Anzeigen der Details einer S3-Access-Grants-Instance

```
aws s3control get-access-grants-instance \  
--account-id 111122223333 \  
--region us-east-2
```

Antwort:

```
{  
  "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default",  
  "AccessGrantsInstanceId": "default",  
  "CreatedAt": "2023-05-31T17:54:07.893000+00:00"  
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [GetAccessGrantsInstance](#)
- [GetAccessGrantsInstanceForPrefix](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details einer S3 Access Grants-Instance mithilfe der AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Abrufen einer S3-Access-Grants-Instance

```
public void getAccessGrantsInstance() {
    GetAccessGrantsInstanceRequest getRequest = GetAccessGrantsInstanceRequest.builder()
        .accountId("111122223333")
        .build();
    GetAccessGrantsInstanceResponse getResponse =
        s3Control.getAccessGrantsInstance(getRequest);
    LOGGER.info("GetAccessGrantsInstanceResponse: " + getResponse);
}
```

Antwort:

```
GetAccessGrantsInstanceResponse(
    AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,
    CreatedAt=2023-06-07T01:46:20.507Z)
```

Listen Sie Ihre S3 Access Grants-Instanzen auf

Sie können Ihre S3 Access Grants-Instanzen auflisten, einschließlich der Instanzen, die mit Ihnen geteilt wurden, über AWS Resource Access Manager (AWS RAM).

Sie können Ihre S3 Access Grants-Instances mithilfe der Amazon S3 S3-Konsole auflisten, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

Um Ihre S3 Access Grants-Instances aufzulisten

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

4. Auf der Seite S3 Access Grants werden Ihre S3-Access-Grants-Instances und alle kontoübergreifenden Instances aufgeführt, die mit Ihrem Konto geteilt wurden. Um die Details einer Instance anzuzeigen, wählen Sie Details anzeigen aus.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Auflisten aller S3-Access-Grants-Instances für ein Konto

Diese Aktion listet die S3-Access-Grants-Instances für ein Konto auf. Sie können nur eine S3 Access Grants-Instanz pro Person haben AWS-Region. Diese Aktion listet auch andere kontoübergreifende S3 Access Grants-Instanzen auf, auf die Ihr Konto Zugriff hat.

```
aws s3control list-access-grants-instances \  
  --account-id 111122223333 \  
  --region us-east-2
```

Antwort:

```
{  
  "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default",  
  "AccessGrantsInstanceId": "default",  
  "CreatedAt": "2023-05-31T17:54:07.893000+00:00"  
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [ListAccessGrantsInstances](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details einer S3 Access Grants-Instance mithilfe der AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Auflisten aller S3-Access-Grants-Instances für ein Konto

Diese Aktion listet die S3-Access-Grants-Instances für ein Konto auf. Es kann pro Region nur eine S3-Access-Grants-Instance geben. Diese Aktion kann auch weitere kontoübergreifende S3-Access-Grants-Instances auflisten, auf die Ihr Konto Zugriff hat.

```
public void listAccessGrantsInstances() {
    ListAccessGrantsInstancesRequest listRequest =
        ListAccessGrantsInstancesRequest.builder()
            .accountId("111122223333")
            .build();
    ListAccessGrantsInstancesResponse listResponse =
        s3Control.listAccessGrantsInstances(listRequest);
    LOGGER.info("ListAccessGrantsInstancesResponse: " + listResponse);
}
```

Antwort:

```
ListAccessGrantsInstancesResponse(
    AccessGrantsInstancesList=[
    ListAccessGrantsInstanceEntry(
        AccessGrantsInstanceId=default,
        AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,
        CreatedAt=2023-06-07T04:28:11.728Z
    )
    ]
)
```

Ordnen Sie Ihre IAM Identity Center-Instanz zu oder trennen Sie sie

In Amazon S3 Access Grants können Sie die verknüpfen AWS IAM Identity Center Instanz Ihres Corporate Identity-Verzeichnisses mit einer S3 Access Grants-Instance. Danach können Sie

Zugriffsberechtigungen für die Benutzer und Gruppen Ihres Unternehmensverzeichnisses erstellen, zusätzlich zu AWS Identity and Access Management (IAM) Benutzer und Rollen.

Wenn Sie keine Zugriffsberechtigungen mehr für die Benutzer und Gruppen Ihres Unternehmensverzeichnisses erstellen möchten, können Sie Ihre IAM Identity Center-Instanz von Ihrer S3 Access Grants-Instanz trennen.

Sie können eine IAM Identity Center-Instanz mithilfe der Amazon S3 S3-Konsole zuordnen oder die Zuordnung aufheben, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

Bevor Sie Ihre IAM Identity Center-Instanz mit Ihrer S3 Access Grants-Instanz verknüpfen, müssen Sie Ihr Corporate Identity-Verzeichnis zu IAM Identity Center hinzufügen. Weitere Informationen finden Sie unter [the section called “S3 Access Grants und Unternehmensverzeichnisidentitäten”](#).

Um eine IAM Identity Center-Instanz mit einer S3 Access Grants-Instanz zu verknüpfen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instanz aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instanz aus.
5. Wählen Sie auf der Detailseite im Abschnitt IAM Identity Center entweder eine Identity Center-Instanz hinzufügen oder die Registrierung einer bereits verknüpften IAM Identity Center-Instanz aufheben.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example — Ordnen Sie eine IAM Identity Center-Instanz einer S3 Access Grants-Instanz zu

```
aws s3control associate-access-grants-identity-center \
```

```
--account-id 111122223333 \  
--identity-center-arn arn:aws:sso:::instance/ssoins-1234a567bb89012c \  
--profile access-grants-profile \  
--region eu-central-1  
  
// No response body
```

Example — Trennen Sie eine IAM Identity Center-Instanz von einer S3 Access Grants-Instanz

```
aws s3control dissociate-access-grants-identity-center \  
--account-id 111122223333 \  
--profile access-grants-profile \  
--region eu-central-1  
  
// No response body
```

Verwenden von REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung der Verknüpfung zwischen einer IAM Identity Center-Instanz und einer S3 Access Grants-Instanz finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [AssociateAccessGrantsIdentityCenter](#)
- [DissociateAccessGrantsIdentityCenter](#)

Löschen einer S3-Access-Grants-Instanz

Sie können eine Amazon S3 Access Grants-Instanz aus einer AWS-Region in Ihrem Konto löschen. Bevor Sie jedoch eine S3-Access-Grants-Instanz löschen können, müssen Sie Folgendes ausführen:

- Löschen Sie alle Ressourcen innerhalb der S3-Access-Grants-Instanz, einschließlich aller Gewährungen und Speicherorte. Weitere Informationen finden Sie unter [Eine Gewährung löschen](#) und [Einen Speicherort löschen](#).
- Wenn Sie eine Instanz mit Ihrer AWS IAM Identity Center Instanz verknüpft haben, müssen Sie die IAM Identity Center-Instanz trennen. Weitere Informationen finden Sie unter [Ihre IAM Identity Center-Instanz zuordnen oder deren Zuordnung aufheben](#).

⚠ Important

Wenn Sie eine S3-Access-Grants-Instance löschen, wird sie dauerhaft gelöscht. Dieser Vorgang kann nicht rückgängig gemacht werden. Alle Gewährungsempfänger, die über die Gewähungen in dieser S3-Access-Grants-Instance Zugriff erhalten haben, verlieren den Zugriff auf Ihre S3-Daten.

Sie können eine S3 Access Grants-Instance mithilfe der Amazon S3 S3-Konsole löschen, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

So löschen Sie eine S3-Access-Grants-Instance

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewähungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie oben rechts auf der Instance-Detailseite Instance löschen aus.
6. Wählen Sie im anschließend angezeigten Dialogfeld Löschen aus. Diese Aktion kann nicht mehr rückgängig gemacht werden.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

ℹ Note

Bevor Sie eine S3-Access-Grants-Instance löschen können, müssen Sie zunächst alle Gewähungen und Speicherorte löschen, die in der S3-Access-Grants-Instance erstellt

wurden. Wenn Sie Ihrer S3 Access Grants-Instanz eine IAM Identity Center-Instanz zugeordnet haben, müssen Sie sie zuerst trennen.

Example – Löschen einer S3-Access-Grants-Instance

```
aws s3control delete-access-grants-instance \  
--account-id 111122223333 \  
--profile access-grants-profile \  
--region us-east-2 \  
--endpoint-url https://s3-control.us-east-2.amazonaws.com \  
  
// No response body
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für das Löschen einer S3 Access Grants-Instance finden Sie unter [DeleteAccessGrantsInstance](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele für das Löschen einer S3 Access Grants-Instance mithilfe von AWS SDKs.

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Note

Bevor Sie eine S3-Access-Grants-Instance löschen können, müssen Sie zunächst alle Gewährungen und Speicherorte löschen, die in der S3-Access-Grants-Instance erstellt wurden. Wenn Sie Ihrer S3 Access Grants-Instanz eine IAM Identity Center-Instanz zugeordnet haben, müssen Sie sie zuerst trennen.

Example – Löschen einer S3-Access-Grants-Instance

```
public void deleteAccessGrantsInstance() {
```

```
DeleteAccessGrantsInstanceRequest deleteRequest =
    DeleteAccessGrantsInstanceRequest.builder()
        .accountId("111122223333")
        .build();
DeleteAccessGrantsInstanceResponse deleteResponse =
    s3Control.deleteAccessGrantsInstance(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResponse: " + deleteResponse);
}
```

Mit S3 Access Grants-Standorten arbeiten

Nachdem Sie [eine Amazon S3 Access Grants-Instance](#) in einem AWS-Region In Ihrem Konto registrieren Sie einen S3-Standort in dieser Instance. Ein S3 Access Grants-Standort ordnet den standardmäßigen S3-Standort (`s3://`), einen Bucket oder ein Präfix einem AWS Identity and Access Management (IAM) Rolle. S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an den Empfänger weiterzugeben, der auf diesen bestimmten Standort zugreift. Sie müssen zuerst mindestens einen Standort in Ihrer S3 Access Grants-Instanz registrieren, bevor Sie eine Zugriffsberechtigung erstellen können.

Sie können einen Standort registrieren, die Details eines Standorts anzeigen, einen Standort bearbeiten und einen Standort löschen.

Note

Nachdem Sie den ersten Standort in Ihrer S3 Access Grants-Instanz registriert haben, enthält Ihre Instance immer noch keine individuellen Zugriffsberechtigungen. Informationen zum Erstellen einer Zugriffsberechtigung finden Sie unter [Erstellen von Gewährungen](#).

Themen

- [Registrieren eines Speicherorts](#)
- [Anzeigen der Details eines registrierten Speicherorts](#)
- [Aktualisieren eines registrierten Speicherorts](#)
- [Löschen eines registrierten Speicherorts](#)

Registrieren eines Speicherorts

Nachdem Sie [eine Amazon S3 Access Grants-Instance](#) in einem AWS-Region In Ihrem Konto registrieren Sie einen S3-Standort in dieser Instance. Ein S3 Access Grants-Standort ordnet den standardmäßigen S3-Standort (`s3://`), einen Bucket oder ein Präfix einem AWS Identity and Access Management (IAM) Rolle. S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an den Empfänger weiterzugeben, der auf diesen bestimmten Standort zugreift. Sie müssen zuerst mindestens einen Standort in Ihrer S3 Access Grants-Instanz registrieren, bevor Sie eine Zugriffsberechtigung erstellen können.

Empfohlener Anwendungsfall

Wir empfehlen, den Standardstandort (`s3://`) zu registrieren und ihn einer IAM Rolle zuzuordnen. Der Standort am standardmäßigen S3-Pfad (`s3://`) deckt den Zugriff auf all Ihre darin enthaltenen S3-Buckets ab AWS-Region Ihres Kontos. Wenn Sie eine Zugriffsberechtigung erstellen, können Sie den Umfang der Gewährung auf einen Bereich, ein Präfix oder ein Objekt innerhalb des Standardverzeichnis einschränken.

Komplexe Anwendungsfälle für die Zugriffsverwaltung

Bei komplexeren Anwendungsfällen für die Zugriffsverwaltung müssen Sie möglicherweise mehr als den Standardspeicherort registrieren. Einige Beispiele für solche Anwendungsfälle sind:

- Nehmen wir an, dass *amzn-s3-demo-bucket* ist ein registrierter Standort in Ihrer S3 Access Grants-Instance, dem eine IAM Rolle zugeordnet ist, aber dieser IAM Rolle wird der Zugriff auf ein bestimmtes Präfix innerhalb des Buckets verweigert. In diesem Fall können Sie das Präfix, auf das die IAM Rolle keinen Zugriff hat, als separaten Standort registrieren und diesen Standort einer anderen IAM Rolle mit dem erforderlichen Zugriff zuordnen.
- Angenommen, Sie möchten Zuweisungen einrichten, die den Zugriff nur auf Benutzer innerhalb eines Virtual Private Cloud (VPC) -Endpunkts beschränken. In diesem Fall können Sie einen Standort für einen Bucket registrieren, in dem die IAM Rolle den Zugriff auf den VPC Endpunkt einschränkt. Wenn später ein Empfänger S3 Access Grants um Anmeldeinformationen bittet, übernimmt S3 Access Grants die IAM Rolle des Standorts, die temporären Anmeldeinformationen zu verkaufen. Diese Anmeldeinformationen verweigern den Zugriff auf den jeweiligen Bucket, sofern sich der Anrufer nicht innerhalb des Endpunkts befindet. VPC Diese Ablehnungsberechtigung wird zusätzlich zu der regulären READ oder in der Gewährung angegebenen READWRITE Berechtigung angewendet. WRITE

Wenn Sie einen Standort registrieren, müssen Sie auch die IAM Rolle angeben, die S3 Access Grants beim Verteilen temporärer Anmeldeinformationen und beim Gültigkeitsbereich der Berechtigungen für einen bestimmten Zuschuss übernimmt.

Wenn Ihr Anwendungsfall erfordert, dass Sie mehrere Standorte in Ihrer S3 Access Grants-Instanz registrieren, können Sie einen der folgenden Standorte registrieren:

S3 URI	IAMRolle	Beschreibung
<code>s3://</code>	<i>Default-IAM-role</i>	Der Standardspeicherort, <code>s3://</code> , umfasst alle Buckets in AWS-Region.
<code>s3://amzn-s3-demo-bucket1 /</code>	<i>IAM-role-For-bucket</i>	Dieser Speicherort enthält alle Objekte im angegebenen Bucket.
<code>s3://amzn-s3-demo-bucket1 /prefix-name</code>	<i>IAM-role-For-prefix</i>	Dieser Speicherort umfasst alle Objekte im Bucket mit einem Objektschlüsselnamen, der mit diesem Präfix beginnt.

Bevor Sie einen bestimmten Bucket oder ein bestimmtes Präfix registrieren können, stellen Sie sicher, dass Sie Folgendes tun:

- Erstellen Sie einen oder mehrere Buckets mit den Daten, auf die Sie Zugriff gewähren möchten. Diese Buckets müssen sich im selben Ordner befinden AWS-Region als Ihre S3 Access Grants-Instanz. Weitere Informationen finden Sie unter [Bucket erstellen](#).

Das Hinzufügen eines Präfixes ist ein optionaler Schritt. Präfixe sind Zeichenketten am Anfang eines Objektschlüsselnamens. Sie können sie sowohl zur Organisation von Objekten in Ihrem Bucket als auch zur Zugriffsverwaltung verwenden. Informationen zum Hinzufügen eines Präfixes zu einem Bucket finden Sie unter [Erstellen von Objektschlüsselnamen](#).

- Erstellen Sie eine IAM Rolle, die berechtigt ist, auf Ihre S3-Daten zuzugreifen, in AWS-Region. Weitere Informationen finden Sie unter [IAMRollen erstellen](#) in AWS IAM Identity Center Benutzerleitfaden.
- Geben Sie in der IAM Rollenvertrauensrichtlinie dem Dienst S3 Access Grants (`access-grants.s3.amazonaws.com`) den Hauptzugriff auf die IAM Rolle, die Sie erstellt haben. Dazu können Sie eine JSON Datei erstellen, die die folgenden Anweisungen enthält. Informationen

zum Hinzufügen der Vertrauensrichtlinie zu Ihrem Konto finden Sie unter [Eine Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien erstellen](#).

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity", "sts:SetContext"],
      "Effect": "Allow",
      "Principal": {"Service": "access-grants.s3.amazonaws.com"},
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId",
          "aws:SourceArn": "arn:aws:s3:region:accountId:access-grants/default"
        }
      }
    }
  ]
  //Optionally, for an IAM Identity Center use case, add:
  "ForAnyValue:StringEquals": {
    "aws:RequestContextProvider": "arn:aws:iam::aws:contextProvider/IdentityCenter"
  }
}
```

- Erstellen Sie eine IAM Richtlinie, um der von Ihnen erstellten IAM Rolle Amazon S3 S3-Berechtigungen zuzuweisen. Schauen Sie sich die folgende Beispieldatei `iam-policy.json` an und ersetzen Sie die *user input placeholders* durch eigene Daten.

Note

- Wenn Sie serverseitige Verschlüsselung verwenden mit AWS Key Management Service (AWS KMS) Schlüssel zur Verschlüsselung Ihrer Daten. Das folgende Beispiel enthält die erforderlichen AWS KMS Berechtigungen für die IAM Rolle in der Richtlinie. Wenn Sie diese Funktion nicht verwenden, können Sie diese Berechtigungen aus Ihrer IAM Richtlinie entfernen.
- Sie können die IAM Rolle nur dann auf den Zugriff auf S3-Daten beschränken, wenn die Anmeldeinformationen von S3 Access Grants vergeben werden. Dieses

Beispiel zeigt Ihnen, wie Sie eine Condition Anweisung für eine bestimmte S3 Access Grants-Instanz hinzufügen. Um dies zu verwenden, ersetzen Sie die S3 Access Grants-Instanz ARN in der Condition Anweisung durch Ihre S3 Access Grants-Instanz ARN, die das folgende Format hat:

```
arn:aws:s3:region:accountId:access-grants/default
```

iam-policy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ObjectLevelReadPermissions",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectVersionAcl",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
          "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-grants/default"]
        }
      }
    },
    {
      "Sid": "ObjectLevelWritePermissions",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl",
        "s3:DeleteObject",

```

```

        "s3:DeleteObjectVersion",
        "s3:AbortMultipartUpload"
    ],
    "Resource": [
        "arn:aws:s3:::*"
    ],
    "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
            "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-Region:accountId:access-grants/default"]
        }
    }
},
{
    "Sid": "BucketLevelReadPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::*"
    ],
    "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
            "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-Region:accountId:access-grants/default"]
        }
    }
},
//Optionally add the following section if you use SSE-KMS encryption
{
    "Sid": "KMSPermissions",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "*"
    ]
}
]

```


}

Sie können einen Standort in Ihrer S3 Access Grants-Instance registrieren, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API oder der AWS SDKs.

Note

Nachdem Sie den ersten Standort in Ihrer S3 Access Grants-Instance registriert haben, enthält Ihre Instance immer noch keine individuellen Zugriffsberechtigungen. Informationen zum Erstellen einer Zugriffsberechtigung finden Sie unter [Erstellen von Gewährungen](#).

Verwenden der S3-Konsole

Bevor Sie mit S3 Access Grants Zugriff auf Ihre S3-Daten gewähren können, müssen Sie über mindestens einen registrierten Speicherort verfügen.

So registrieren Sie einen Speicherort in Ihrer S3-Access-Grants-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

Wenn Sie die S3-Access-Grants-Instance zum ersten Mal verwenden, müssen Sie [Schritt 1 – Erstellen einer S3-Access-Grants-Instance](#) abgeschlossen haben und zu Schritt 2 des Assistenten Access-Grants-Instance einrichten gewechselt sein. Wenn Sie bereits über eine S3-Access-Grants-Instance verfügen, wählen Sie Details anzeigen und dann auf der Registerkarte Speicherorte die Option Speicherort registrieren aus.

- a. Wählen Sie für den Bereich Standort die Option S3 durchsuchen oder geben Sie den URI S3-Pfad zu dem Standort ein, den Sie registrieren möchten. Informationen zu URI S3-Formaten finden Sie in der Tabelle mit den [Speicherformaten](#). Nachdem Sie einen eingegeben habenURI, können Sie „Ansicht“ wählen, um den Speicherort zu durchsuchen.
- b. Wählen Sie für die IAMRolle eine der folgenden Optionen aus:

- Wählen Sie aus vorhandenen IAM Rollen

Wählen Sie eine IAM Rolle aus der Drop-down-Liste aus. Wählen Sie nach der Auswahl einer Rolle Anzeigen aus, um sicherzustellen, dass diese Rolle die notwendigen Berechtigungen für die Verwaltung des Speicherorts besitzt, den Sie registrieren. Stellen Sie insbesondere sicher, dass diese Rolle S3 Access Grants die Berechtigungen `sts:AssumeRole` und `sts:SetSourceIdentity` gewährt.

- Geben Sie die Rolle ein IAM ARN

Navigieren Sie zur [IAMKonsole](#). Kopieren Sie den Amazon-Ressourcennamen (ARN) der IAM Rolle und fügen Sie ihn in dieses Feld ein.

c. Wählen Sie Weiter oder Speicherort registrieren aus, um dies abzuschließen.

4. Fehlerbehebung

Speicherort kann nicht registriert werden

- Der Speicherort ist möglicherweise bereits registriert.

Möglicherweise besitzen Sie nicht die Berechtigung `s3:CreateAccessGrantsLocation` für die Registrierung von Speicherorten. Nehmen Sie Kontakt mit Ihrem Kontoadministrator auf.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Sie können in Ihrer S3-Access-Grants-Instance den Standardspeicherort (`s3://`) oder einen benutzerdefinierten Speicherort registrieren. Stellen Sie sicher, dass Sie zuerst eine IAM Rolle mit Prinzipalzugriff auf den Standort erstellen und dann sicherstellen, dass Sie S3 Access Grants die Erlaubnis erteilen, diese Rolle anzunehmen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example Erstellen einer Ressourcenrichtlinie

Erstellen Sie eine Richtlinie, die es S3 Access Grants ermöglicht, die IAM Rolle zu übernehmen. Dazu können Sie eine JSON Datei erstellen, die die folgenden Anweisungen enthält. Informationen

zum Hinzufügen der Ressourcenrichtlinie zu Ihrem Konto finden Sie unter [Erstellen und Anfügen Ihrer ersten vom Kunden verwalteten Richtlinie](#).

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity"],
      "Effect": "Allow",
      "Principal": {"Service": "access-grants.s3.amazonaws.com"}
    }
  ]
}
```

Example Erstellen der Rolle

Führen Sie den folgenden IAM Befehl aus, um die Rolle zu erstellen.

```
aws iam create-role --role-name accessGrantsTestRole \
  --region us-east-2 \
  --assume-role-policy-document file://TestRolePolicy.json
```

Wenn Sie den Befehl `create-role` ausführen, wird die Richtlinie zurückgegeben:

```
{
  "Role": {
    "Path": "/",
    "RoleName": "accessGrantsTestRole",
    "RoleId": "AROASRDGX4WM4GH55GIDA",
    "Arn": "arn:aws:iam::111122223333:role/accessGrantsTestRole",
    "CreateDate": "2023-05-31T18:11:06+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "Stmt1685556427189",
          "Action": [
            "sts:AssumeRole",
            "sts:SetSourceIdentity"
          ]
        }
      ]
    }
  }
}
```

```
    ],  
    "Effect": "Allow",  
    "Principal": {  
        "Service": "access-grants.s3.amazonaws.com"  
    }  
  }  
]  
}  
}
```

Example

Erstellen Sie eine IAM Richtlinie, um Amazon S3 S3-Berechtigungen an die IAM Rolle anzuhängen. Schauen Sie sich die folgende Beispieldatei `iam-policy.json` an und ersetzen Sie die *user input placeholders* durch eigene Daten.

Note

Wenn Sie serverseitige Verschlüsselung verwenden mit AWS Key Management Service (AWS KMS) Schlüssel zum Verschlüsseln Ihrer Daten. Im folgenden Beispiel werden die erforderlichen Schlüssel hinzugefügt AWS KMS Berechtigungen für die IAM Rolle in der Richtlinie. Wenn Sie diese Funktion nicht verwenden, können Sie diese Berechtigungen aus Ihrer IAM Richtlinie entfernen.

Um sicherzustellen, dass die IAM Rolle nur für den Zugriff auf Daten in S3 verwendet werden kann, wenn die Anmeldeinformationen von S3 Access Grants ausgegeben werden, zeigt Ihnen dieses Beispiel, wie Sie Ihrer IAM Richtlinie eine Condition Anweisung hinzufügen, die die S3 Access Grants-Instanz (`s3:AccessGrantsInstance: InstanceArn`) angibt. Wenn Sie das folgende Beispiel verwenden, ersetzen Sie die *user input placeholders* durch eigene Daten.

iam-policy.json

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "ObjectLevelReadPermissions",
```

```

    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectVersionAcl",
        "s3:ListMultipartUploadParts"
    ],
    "Resource": [
        "arn:aws:s3:::*"
    ],
    "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
            "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-
grants/default"]
        }
    }
},
{
    "Sid": "ObjectLevelWritePermissions",
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl",
        "s3>DeleteObject",
        "s3>DeleteObjectVersion",
        "s3:AbortMultipartUpload"
    ],
    "Resource": [
        "arn:aws:s3:::*"
    ],
    "Condition": {
        "StringEquals": { "aws:ResourceAccount": "accountId" },
        "ArnEquals": {
            "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-Region:accountId:access-
grants/default"]
        }
    }
},
{
    "Sid": "BucketLevelReadPermissions",
    "Effect": "Allow",

```

```

    "Action":[
      "s3:ListBucket"
    ],
    "Resource":[
      "arn:aws:s3:::*"
    ],
    "Condition":{
      "StringEquals": { "aws:ResourceAccount": "accountId" },
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-Region:accountId:access-
grants/default"]
      }
    }
  },
  {
    "Sid": "KMSPermissions",
    "Effect":"Allow",
    "Action":[
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource":[
      "*"
    ]
  }
]
}

```

Example

Führen Sie den folgenden Befehl aus:

```

aws iam put-role-policy \
--role-name accessGrantsTestRole \
--policy-name accessGrantsTestRole \
--policy-document file://iam-policy.json

```

Example Registrieren des Standardspeicherorts

```

aws s3control create-access-grants-location \
--account-id 111122223333 \
--location-scope s3:// \
--iam-role-arn arn:aws:iam::111122223333:role/accessGrantsTestRole

```

Antwort:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
  "AccessGrantsLocationId": "default",
  "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
  "LocationScope": "s3://",
  "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Example Registrieren eines benutzerdefinierten Speicherorts

```
aws s3control create-access-grants-location \
--account-id 111122223333 \
--location-scope s3://DOC-BUCKET-EXAMPLE/ \
--iam-role-arn arn:aws:iam::123456789012:role/accessGrantsTestRole
```

Antwort:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
  "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
  "AccessGrantsLocationArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
  "LocationScope": "s3://DOC-BUCKET-EXAMPLE/",
  "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [CreateAccessGrantsLocation](#)
- [DeleteAccessGrantsLocation](#)
- [GetAccessGrantsLocation](#)
- [ListAccessGrantsLocations](#)
- [UpdateAccessGrantsLocation](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele für die Registrierung von Standorten mithilfe der AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Sie können in Ihrer S3-Access-Grants-Instance den Standardspeicherort (`s3://`) oder einen benutzerdefinierten Speicherort registrieren. Stellen Sie sicher, dass Sie zuerst eine IAM Rolle mit Hauptzugriff auf den Standort erstellen und dann sicherstellen, dass Sie S3 Access Grants die Berechtigung erteilen, diese Rolle anzunehmen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example Registrieren eines Standardspeicherorts

Anfrage:

```
public void createAccessGrantsLocation() {
    CreateAccessGrantsLocationRequest createRequest =
        CreateAccessGrantsLocationRequest.builder()
            .accountId("111122223333")
            .locationScope("s3://")
            .iamRoleArn("arn:aws:iam::123456789012:role/accessGrantsTestRole")
            .build();
    CreateAccessGrantsLocationResponse createResponse =
        s3Control.createAccessGrantsLocation(createRequest);
    LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Antwort:

```
CreateAccessGrantsLocationResponse(
    CreatedAt=2023-06-07T04:35:11.027Z,
    AccessGrantsLocationId=default,
    AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
    location/default,
    LocationScope=s3://,
    IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
```


)

Example Registrieren eines benutzerdefinierten Speicherorts

Anfrage:

```
public void createAccessGrantsLocation() {
    CreateAccessGrantsLocationRequest createRequest =
        CreateAccessGrantsLocationRequest.builder()
            .accountId("111122223333")
            .locationScope("s3://DOC-BUCKET-EXAMPLE/")
            .iamRoleArn("arn:aws:iam::111122223333:role/accessGrantsTestRole")
            .build();
    CreateAccessGrantsLocationResponse createResponse =
        s3Control.createAccessGrantsLocation(createRequest);
    LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Antwort:

```
CreateAccessGrantsLocationResponse(
    CreatedAt=2023-06-07T04:35:10.027Z,
    AccessGrantsLocationId=18cfe6fb-eb5a-4ac5-aba9-8d79f04c2012,
    AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
    location/18cfe6fb-eb5a-4ac5-aba9-8d79f04c2666,
    LocationScope= s3://test-bucket-access-grants-user123/,
    IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Anzeigen der Details eines registrierten Speicherorts

Sie können die Details eines Standorts abrufen, der in Ihrer S3 Access Grants-Instance registriert ist, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

So zeigen Sie die die in Ihrer S3-Access-Grants-Instance gespeicherten Speicherorte an

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite für die Instance die Registerkarte Speicherorte aus.
6. Suchen Sie den registrierten Speicherort, den Sie anzeigen möchten. Verwenden Sie das Suchfeld, um die Liste der registrierten Speicherorte zu filtern.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Abrufen der Details eines registrierten Speicherorts

```
aws s3control get-access-grants-location \  
--account-id 111122223333 \  
--access-grants-location-id default
```

Antwort:

```
{  
  "CreatedAt": "2023-05-31T18:23:48.107000+00:00",  
  "AccessGrantsLocationId": "default",  
  "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default/location/default",  
  "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"  
}
```

Example – Auflisten aller in einer S3-Access-Grants-Instance registrierten Speicherorte

Um die Ergebnisse auf ein S3-Präfix oder einen S3-Bucket einzuschränken, können Sie optional den Parameter `--location-scope s3://bucket-and-or-prefix` verwenden.

```
aws s3control list-access-grants-locations \  
--account-id 111122223333 \  
--location-scope s3://bucket-and-or-prefix
```

```
--region us-east-2
```

Antwort:

```
{"AccessGrantsLocationsList": [  
  {  
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",  
    "AccessGrantsLocationId": "default",  
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default/location/default",  
    "LocationScope": "s3://"   
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"  
  },  
  {  
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",  
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",  
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default/location/635f1139-1af2-4e43-8131-a4de006eb888",  
    "LocationScope": "s3://amzn-s3-demo-bucket/prefixA*",  
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"  
  }  
]  
}
```

Mit dem REST API

Informationen zum Amazon S3 REST API S3-Support zum Abrufen der Details eines registrierten Standorts oder zum Auflisten aller Standorte, die bei einer S3 Access Grants-Instance registriert sind, finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [GetAccessGrantsLocation](#)
- [ListAccessGrantsLocations](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details eines registrierten Standorts abrufen oder alle registrierten Standorte in einer S3 Access Grants-Instance auflisten können, indem Sie AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Abrufen der Details eines registrierten Speicherorts

```
public void getAccessGrantsLocation() {
    GetAccessGrantsLocationRequest getAccessGrantsLocationRequest =
        GetAccessGrantsLocationRequest.builder()
            .accountId("111122223333")
            .accessGrantsLocationId("default")
            .build();
    GetAccessGrantsLocationResponse getAccessGrantsLocationResponse =
        s3Control.getAccessGrantsLocation(getAccessGrantsLocationRequest);
    LOGGER.info("GetAccessGrantsLocationResponse: " + getAccessGrantsLocationResponse);
}
```

Antwort:

```
GetAccessGrantsLocationResponse(
    CreatedAt=2023-06-07T04:35:10.027Z,
    AccessGrantsLocationId=default,
    AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
    LocationScope= s3://,
    IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Example – Auflisten aller in einer S3-Access-Grants-Instance registrierten Speicherorte

Um die Ergebnisse auf ein S3-Präfix oder einen S3-Bucket zu beschränken, können Sie optional ein S3URI, z. B. `s3://bucket-and-or-prefix`, im `LocationScope` Parameter übergeben.

```
public void listAccessGrantsLocations() {

    ListAccessGrantsLocationsRequest listRequest =
        ListAccessGrantsLocationsRequest.builder()
            .accountId("111122223333")
            .build();

    ListAccessGrantsLocationsResponse listResponse =
        s3Control.listAccessGrantsLocations(listRequest);
    LOGGER.info("ListAccessGrantsLocationsResponse: " + listResponse);
}
```

Antwort:

```
ListAccessGrantsLocationsResponse(  
  AccessGrantsLocationsList=[  
    ListAccessGrantsLocationsEntry(  
      CreatedAt=2023-06-07T04:35:11.027Z,  
      AccessGrantsLocationId=default,  
      AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/  
location/default,  
      LocationScope=s3://,  
      IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole  
    ),  
    ListAccessGrantsLocationsEntry(  
      CreatedAt=2023-06-07T04:35:10.027Z,  
      AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb456,  
      AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/  
location/635f1139-1af2-4e43-8131-a4de006eb888,  
      LocationScope=s3://amzn-s3-demo-bucket/prefixA*,  
      IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole  
    )  
  ]  
)
```

Aktualisieren eines registrierten Speicherorts

Sie können die aktualisieren AWS Identity and Access Management (IAM) Rolle eines Standorts, der in Ihrer Amazon S3 Access Grants-Instance registriert ist. Stellen Sie für jede neue IAM Rolle, die Sie zur Registrierung eines Standorts in S3 Access Grants verwenden, sicher, dass Sie dem Dienstprinzipal (`access-grants.s3.amazonaws.com`) von S3 Access Grants Zugriff auf diese Rolle gewähren. Fügen Sie dazu einen Eintrag für die neue IAM Rolle in derselben JSON Vertrauensrichtliniendatei hinzu, die Sie bei der ersten [Registrierung des Standorts](#) verwendet haben.

Sie können einen Standort in Ihrer S3 Access Grants-Instance aktualisieren, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

Um die IAM Rolle eines Standorts zu aktualisieren, der bei Ihrer S3 Access Grants-Instance registriert ist

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite für die Instance die Registerkarte Speicherorte aus.
6. Suchen Sie den Speicherort aus, den Sie aktualisieren möchten. Verwenden Sie das Suchfeld, um die Liste der Speicherorte zu filtern.
7. Wählen Sie die Optionsschaltfläche neben dem registrierten Speicherort aus, den Sie aktualisieren möchten.
8. Aktualisieren Sie die IAM Rolle und wählen Sie dann Änderungen speichern.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example — Aktualisieren Sie die IAM Rolle eines registrierten Standorts

```
aws s3control update-access-grants-location \  
--account-id 111122223333 \  
--access-grants-location-id 635f1139-1af2-4e43-8131-a4de006eb999 \  
--iam-role-arn arn:aws:iam::777788889999:role/accessGrantsTestRole
```

Antwort:

```
{  
  "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
```

```
"AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb999",
"AccessGrantsLocationArn": "arn:aws:s3:us-east-2:777788889999:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
"LocationScope": "s3://amzn-s3-demo-bucket/prefixB*",
"IAMRoleArn": "arn:aws:iam::777788889999:role/accessGrantsTestRole"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Aktualisierung eines Standorts in einer S3 Access Grants-Instance finden Sie unter [UpdateAccessGrantsLocation](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele für die Aktualisierung der IAM Rolle eines registrierten Standorts mithilfe der AWS SDKs.

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example — Aktualisieren Sie die IAM Rolle eines registrierten Standorts

```
public void updateAccessGrantsLocation() {
    UpdateAccessGrantsLocationRequest updateRequest =
        UpdateAccessGrantsLocationRequest.builder()
            .accountId("111122223333")
            .accessGrantsLocationId("635f1139-1af2-4e43-8131-a4de006eb999")
            .iamRoleArn("arn:aws:iam::777788889999:role/accessGrantsTestRole")
            .build();
    UpdateAccessGrantsLocationResponse updateResponse =
        s3Control.updateAccessGrantsLocation(updateRequest);
    LOGGER.info("UpdateAccessGrantsLocationResponse: " + updateResponse);
}
```

Antwort:

```
UpdateAccessGrantsLocationResponse(
    CreatedAt=2023-06-07T04:35:10.027Z,
```

```
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb999,  
AccessGrantsLocationArn=arn:aws:s3:us-east-2:777788889999:access-grants/default/  
location/635f1139-1af2-4e43-8131-a4de006eb888,  
LocationScope=s3://amzn-s3-demo-bucket/prefixB*,  
IAMRoleArn=arn:aws:iam::777788889999:role/accessGrantsTestRole  
)
```

Löschen eines registrierten Speicherorts

Sie können eine Speicherortregistrierung aus einer Amazon-S3-Access-Grants-Instance löschen. Durch das Löschen des Speicherorts wird dessen Registrierung in der S3-Access-Grants-Instance aufgehoben.

Bevor Sie eine Speicherortregistrierung aus einer S3-Access-Grants-Instance entfernen können, müssen Sie alle mit diesem Speicherort verknüpften Gewährungen löschen. Informationen zum Löschen von Gewährungen finden Sie unter [Löschen einer Gewährung](#).

Sie können einen Standort in Ihrer S3 Access Grants-Instance löschen, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

So löschen Sie eine Speicherortregistrierung aus Ihrer Amazon-S3-Access-Grants-Instance

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite für die Instance die Registerkarte Speicherorte aus.
6. Suchen Sie den Speicherort aus, den Sie aktualisieren möchten. Verwenden Sie das Suchfeld, um die Liste der Speicherorte zu filtern.
7. Wählen Sie die Optionsschaltfläche neben dem registrierten Speicherort aus, den Sie löschen möchten.
8. Wählen Sie Deregister.

9. Anschließend wird ein Dialogfeld angezeigt, in dem Sie gewarnt werden, dass diese Aktion nicht rückgängig gemacht werden kann. Um den Speicherort zu löschen, wählen Sie Registrierung aufheben aus.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Löschen einer Speicherortregistrierung

```
aws s3control delete-access-grants-location \  
--account-id 111122223333 \  
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
// No response body
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für das Löschen eines Standorts aus einer S3 Access Grants-Instance finden Sie unter [DeleteAccessGrantsLocation](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Dieser Abschnitt enthält ein Beispiel für das Löschen eines Speicherorts mithilfe von AWS SDKs.

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Löschen einer Speicherortregistrierung

```
public void deleteAccessGrantsLocation() {  
    DeleteAccessGrantsLocationRequest deleteRequest =  
        DeleteAccessGrantsLocationRequest.builder()  
            .accountId("111122223333")
```

```
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
.build();
DeleteAccessGrantsLocationResponse deleteResponse =
    s3Control.deleteAccessGrantsLocation(deleteRequest);
LOGGER.info("DeleteAccessGrantsLocationResponse: " + deleteResponse);
}
```

Antwort:

```
DeleteAccessGrantsLocationResponse()
```

Arbeiten mit Zuschüssen in S3 Access Grants

Eine individuelle Zugriffsberechtigung in einer S3 Access Grants-Instanz ermöglicht eine bestimmte Identität — eine AWS Identity and Access Management (IAM) Principal oder ein Benutzer oder eine Gruppe in einem Unternehmensverzeichnis — um Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist. Ein Standort ordnet Buckets oder Präfixe einer Rolle zu. IAM S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an Empfänger zu verkaufen.

Nachdem Sie [mindestens einen Standort in Ihrer S3 Access Grants-Instanz registriert](#) haben, können Sie eine Zugriffsberechtigung erstellen.

Der Empfänger kann ein IAM Benutzer oder eine Rolle oder ein Verzeichnisbenutzer oder eine Gruppe sein. Ein Verzeichnisbenutzer ist ein Benutzer aus Ihrem Unternehmensverzeichnis oder einer externen Identitätsquelle, den Sie [mit Ihrer S3 Access Grants-Instanz verknüpft](#) haben. Weitere Informationen finden Sie unter [S3 Access Grants und Unternehmensverzeichnisidentitäten](#). Um einen Zuschuss für einen bestimmten Verzeichnisbenutzer oder eine bestimmte Verzeichnisgruppe aus IAM Identity Center zu erstellen, suchen Sie in IAM Identity Center nach dem GUID, anhand dessen IAM Identity Center diesen Benutzer identifiziert, a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 z. B. Weitere Informationen zur Verwendung von IAM Identity Center zum Anzeigen von Benutzerinformationen finden Sie unter [Benutzer- und Gruppenzuweisungen anzeigen](#) in AWS IAM Identity Center Benutzerleitfaden.

Sie können Zugriff auf einen Bucket, ein Präfix oder ein Objekt gewähren. Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, mit der die Objekte in einem Bucket organisiert werden. Dabei kann es sich um eine beliebige Zeichenfolge mit zulässigen Zeichen handeln, z. B. Objektschlüsselnamen in Ihrem Bucket, die mit dem `engineering/` Präfix beginnen.

Themen

- [Erstellen von Gewährungen](#)
- [Anzeigen einer Gewährung](#)
- [Löschen einer Gewährung](#)

Erstellen von Gewährungen

Eine individuelle Zugriffsgewährung in einer S3 Access Grants-Instanz ermöglicht eine bestimmte Identität — eine AWS Identity and Access Management (IAM) Principal oder ein Benutzer oder eine Gruppe in einem Unternehmensverzeichnis — um Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist. Ein Standort ordnet Buckets oder Präfixe einer Rolle zu. IAM S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an Empfänger zu verkaufen.

Nachdem Sie [mindestens einen Standort in Ihrer S3 Access Grants-Instanz registriert](#) haben, können Sie eine Zugriffsberechtigung erstellen.

Der Empfänger kann ein IAM Benutzer oder eine Rolle oder ein Verzeichnisbenutzer oder eine Gruppe sein. Ein Verzeichnisbenutzer ist ein Benutzer aus Ihrem Unternehmensverzeichnis oder einer externen Identitätsquelle, den Sie [mit Ihrer S3 Access Grants-Instanz verknüpft](#) haben. Weitere Informationen finden Sie unter [S3 Access Grants und Unternehmensverzeichnisidentitäten](#). Um einen Zuschuss für einen bestimmten Verzeichnisbenutzer oder eine bestimmte Verzeichnisgruppe aus IAM Identity Center zu erstellen, suchen Sie in IAM Identity Center nach dem GUID, anhand dessen IAM Identity Center diesen Benutzer identifiziert, a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 z. B. Weitere Informationen zur Verwendung von IAM Identity Center zum Anzeigen von Benutzerinformationen finden Sie unter [Benutzer- und Gruppenzuweisungen anzeigen](#) in AWS IAM Identity Center Benutzerleitfaden.

Sie können Zugriff auf einen Bucket, ein Präfix oder ein Objekt gewähren. Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, mit der die Objekte in einem Bucket organisiert werden. Dabei kann es sich um eine beliebige Zeichenfolge mit zulässigen Zeichen handeln, z. B. Objektschlüsselnamen in Ihrem Bucket, die mit dem `engineering/` Präfix beginnen.

Unterpräfix

Wenn Sie Zugriff auf einen registrierten Standort gewähren, können Sie das `Subprefix` Feld verwenden, um den Zugriffsbereich auf eine Teilmenge des Standortbereichs einzuschränken. Wenn

es sich bei dem registrierten Standort, den Sie für die Gewährung auswählen, um den Standard-S3-Pfad (`s3://`) handelt, müssen Sie den Geltungsbereich der Gewährung einschränken. Sie können keine Zugriffsberechtigung für den Standardspeicherort (`s3://`) erstellen, wodurch der Empfänger Zugriff auf jeden Bucket in einer AWS-Region. Stattdessen müssen Sie den Umfang des Zuschusses auf einen der folgenden Bereiche einschränken:

- Ein Eimer: `s3://bucket/*`
- Ein Präfix innerhalb eines Buckets: `s3://bucket/prefix*`
- Ein Präfix innerhalb eines Präfixes: `s3://bucket/prefixA/prefixB*`
- Ein Objekt: `s3://bucket/object-key-name`

Wenn Sie eine Zugriffsberechtigung erstellen, bei der es sich bei dem registrierten Standort um einen Bucket handelt, können Sie eine der folgenden Angaben in das Subprefix-Feld eingeben, um den Umfang der Gewährung einzuschränken:

- Ein Präfix innerhalb des Buckets: `prefix*`
- Ein Präfix innerhalb eines Präfixes: `prefixA/prefixB*`
- Ein Objekt: `/object-key-name`

Nachdem Sie den Zuschuss erstellt haben, wird der Umfang der Gewährung, der in der Amazon S3 S3-Konsole angezeigt wird, `GrantScope`, API oder AWS Command Line Interface (AWS CLI). Die Antwort ist das Ergebnis der Verkettung des Standortpfads mit dem Subprefix. Dieser verkettete Pfad muss mit dem S3-Bucket, Präfix oder Objekt übereinstimmen, dem Sie Zugriff gewähren möchten.

Note

- Wenn Sie eine Zugriffserteilung erstellen müssen, die nur Zugriff auf ein Objekt gewährt, müssen Sie angeben, dass der Gewährungstyp für ein Objekt gilt. Um dies in einem API-Aufruf oder einem CLI-Befehl zu tun, übergeben Sie den `s3PrefixType`-Parameter mit dem Wert `Object`. Wenn Sie in der Amazon S3 S3-Konsole den Grant erstellen, nachdem Sie einen Standort ausgewählt haben, aktivieren Sie unter Grant Scope das Kontrollkästchen `Grant scope is an object`.
- Sie können keine Gewährung für einen Bucket erstellen, der noch nicht vorhanden ist. Sie können jedoch einen Zuschuss für ein Präfix erstellen, das noch nicht existiert.

- Die maximale Anzahl von Zuschüssen, die Sie in Ihrer S3 Access Grants-Instanz erstellen können, finden Sie unter [Limits für S3 Access Grants](#).

Sie können mithilfe der Amazon S3 S3-Konsole eine Zugriffsgewährung erstellen. AWS CLI, der Amazon S3 REST API und AWS SDKs.

Verwenden der S3-Konsole

So erstellen Sie eine Zugriffsgewährung

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

Wenn Sie die S3-Access-Grants-Instance zum ersten Mal verwenden, müssen Sie [Schritt 2 – Registrieren eines Speicherorts](#) abgeschlossen haben und zu Schritt 3 des Assistenten Access-Grants-Instance einrichten gewechselt sein. Wenn Sie bereits über eine S3-Access-Grants-Instance verfügen, wählen Sie Details anzeigen und dann auf der Registerkarte Gewährungen die Option Gewährung erstellen aus.

- a. Wählen Sie im Abschnitt Gewährungsumfang einen registrierten Standort aus oder geben Sie diesen ein.

Wenn Sie den `s3://`-Standardspeicherort ausgewählt haben, können Sie im Feld Unterpräfix den Umfang der Zugriffsgewährung einschränken. Weitere Informationen finden Sie unter [Unterpräfix](#). Wenn Sie lediglich einem Objekt Zugriff gewähren, wählen Sie Gewährungsumfang ist ein Objekt aus.

- b. Wählen Sie unter Berechtigungen und Zugriff die Stufe der Berechtigung aus, Lesen, Schreiben oder beides.

Wählen Sie dann den Empfängertyp aus. Wenn Sie Ihr Unternehmensverzeichnis zu IAM Identity Center hinzugefügt und diese IAM Identity Center-Instance mit Ihrer S3 Access Grants-Instance verknüpft haben, können Sie die Verzeichnisidentität aus IAM Identity Center auswählen. Wenn Sie diese Option wählen, rufen Sie die ID des Benutzers oder der Gruppe aus IAM Identity Center ab und geben Sie sie in diesen Abschnitt ein.

Wenn der Typ des Empfängers ein IAM Benutzer oder eine Rolle ist, wählen Sie IAMPrincipal. Wählen Sie unter IAMPrinzipaltyp die Option Benutzer oder Rolle aus. Wählen Sie dann unter IAMHauptbenutzer entweder aus der Liste aus, oder geben Sie die ID der Identität ein.

- c. Um die S3-Access-Grants-Gewährung zu erstellen, wählen Sie Weiter oder Gewährung erstellen aus.
4. Wenn Weiter oder Gewährung erstellen deaktiviert ist:

Gewährung kann nicht erstellt werden

- Möglicherweise müssen Sie in Ihrer S3-Access-Grants-Instance zuerst [einen Speicherort registrieren](#).
- Möglicherweise besitzen Sie die Berechtigung `s3:CreateAccessGrant` nicht, die für die Erstellung Zugriffsgewährung erforderlich ist. Nehmen Sie Kontakt mit Ihrem Kontoadministrator auf.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Die folgenden Beispiele zeigen, wie Sie eine Zugriffserteilungsanforderung für einen IAM Prinzipal und eine Zugriffserteilungsanforderung für einen Benutzer oder eine Gruppe im Unternehmensverzeichnis erstellen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Note

Wenn Sie eine Zugriffsgewährung erstellen, die Zugriff auf ein einzelnes Objekt gewährt, geben Sie den erforderlichen Parameter `--s3-prefix-type Object` an.

Example Erstellen Sie eine Anfrage zur Zugriffsgewährung für einen IAM Principal

```
aws s3control create-access-grant \
```

```
--account-id 111122223333 \  
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \  
--access-grants-location-configuration S3SubPrefix=prefixB* \  
--permission READ \  
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::123456789012:user/data-  
consumer-3
```

Example Erstellen einer Zugriffsgewährungsantwort

```
{"CreatedAt": "2023-05-31T18:41:34.663000+00:00",  
  "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/  
grant/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "Grantee": {  
    "GranteeType": "IAM",  
    "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"  
  },  
  "AccessGrantsLocationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "AccessGrantsLocationConfiguration": {  
    "S3SubPrefix": "prefixB*"  
  },  
  "GrantScope": "s3://DOC-BUCKET-EXAMPLE/prefix*",  
  "Permission": "READ"  
}
```

Erstellen einer Anforderung für eine Zugriffsgewährung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe

Um eine Zugriffserteilungsanforderung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe zu erstellen, müssen Sie zuerst die GUID für den Verzeichnisbenutzer oder die Verzeichnisgruppe abrufen, indem Sie einen der folgenden Befehle ausführen.

Example Rufen Sie eine GUID für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe ab

Sie finden den Namen GUID eines IAM Identity Center-Benutzers über die IAM Identity Center-Konsole oder mithilfe der AWS CLI or AWS SDKs. Der folgende Befehl listet die Benutzer in der angegebenen IAM Identity Center-Instanz mit ihren Namen und Kennungen auf.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Dieser Befehl listet die Gruppen in der angegebenen IAM Identity Center-Instanz auf.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Erstellen einer Zugriffsgewährung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe

Dieser Befehl ähnelt der Erstellung eines Zuschusses für IAM Benutzer oder Rollen, mit der Ausnahme, dass der Typ des Empfängers `DIRECTORY_USER` oder `DIRECTORY_GROUP` ist und die Empfänger-ID die GUID für den Verzeichnisbenutzer oder die Gruppe ist.

```
aws s3control create-access-grant \  
--account-id 123456789012 \  
--access-grants-location-id default \  
--access-grants-location-configuration S3SubPrefix="amzn-s3-demo-bucket/rafael/*" \  
--permission READWRITE \  
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier=83d43802-00b1-7054-db02-f1d683aacba5 \  

```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung von Zugriffsberechtigungen finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [CreateAccessGrant](#)
- [DeleteAccessGrant](#)
- [GetAccessGrant](#)
- [ListAccessGrants](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie eine Zugriffsberechtigung mithilfe der AWS SDKs.

Java

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Daten.

Note

Wenn Sie eine Zugriffsgewährung erstellen, die Zugriff auf ein einzelnes Objekt gewährt, geben Sie den erforderlichen Parameter `.s3PrefixType(S3PrefixType.Object)` an.

Example Erstellen einer Anforderung für eine Zugriffsgewährung

```
public void createAccessGrant() {
    CreateAccessGrantRequest createRequest = CreateAccessGrantRequest.builder()
        .accountId("111122223333")
        .accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLEeaaaa")
        .permission("READ")
        .accessGrantsLocationConfiguration(AccessGrantsLocationConfiguration.builder().s3SubPrefix("
        .grantee(Grantee.builder().granteeType("IAM").granteeIdentifier("arn:aws:iam::111122223333:u
        data-consumer-3").build())
        .build();
    CreateAccessGrantResponse createResponse =
        s3Control.createAccessGrant(createRequest);
    LOGGER.info("CreateAccessGrantResponse: " + createResponse);
}
```

Example Erstellen einer Zugriffsgewährungsantwort

```
CreateAccessGrantResponse(
    CreatedAt=2023-06-07T05:20:26.330Z,
    AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
    AccessGrantArn=arn:aws:s3:us-east-2:444455556666:access-grants/default/grant/
    a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
    Grantee=Grantee(
        GranteeType=IAM,
        GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
    ),
    AccessGrantsLocationId=a1b2c3d4-5678-90ab-cdef-EXAMPLEeaaaa,
    AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
        S3SubPrefix=prefixB*
    ),
    GrantScope=s3://DOC-BUCKET-EXAMPLE/prefixB,
    Permission=READ
)
```

Anzeigen einer Gewährung

Sie können die Details einer Zugriffserteilung in Ihrer Amazon S3 Access Grants-Instance anzeigen, indem Sie die Amazon S3 S3-Konsole verwenden, die AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

So zeigen Sie die Details einer Zugriffsgewährung an

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite die Registerkarte Gewährungen aus.
6. Suchen Sie im Bereich Gewährungen nach der Zugriffsgewährung, die Sie anzeigen möchten. Sie können die Liste der Gewährungen über das Suchfeld filtern.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example – Abrufen der Details einer Zugriffsgewährung

```
aws s3control get-access-grant \  
--account-id 111122223333 \  
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Antwort:

```
{  
  "CreatedAt": "2023-05-31T18:41:34.663000+00:00",  
  "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```

    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-a1b2c3d4-5678-90ab-cdef-EXAMPLE2222",
    "Grantee": {
      "GranteeType": "IAM",
      "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "Permission": "READ",
    "AccessGrantsLocationId": "12a6710f-5af8-41f5-b035-0bc795bf1a2b",
    "AccessGrantsLocationConfiguration": {
      "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://amzn-s3-demo-bucket/"
  }
}

```

Example – Auflisten aller Zugriffsgewährungen in einer S3-Access-Grants-Instance

Sie können optional die folgenden Parameter verwenden, um die Ergebnisse auf ein S3-Präfix zu beschränken oder AWS Identity and Access Management (IAM) Identität:

- Unterpräfix – `--grant-scope s3://bucket-name/prefix*`
- IAM Identität — `--grantee-type IAM` und `--grantee-identifier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```

aws s3control list-access-grants \
--account-id 111122223333

```

Antwort:

```

{
  "AccessGrantsList": [
    {
      "CreatedAt": "2023-06-14T17:54:46.542000+00:00",
      "AccessGrantId": "dd8dd089-b224-4d82-95f6-975b4185bbaa",
      "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa",
      "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
      },
      "Permission": "READ",
      "AccessGrantsLocationId": "23514a34-ea2e-4ddf-b425-d0d4bfcarda1",
      "GrantScope": "s3://amzn-s3-demo-bucket/prefixA*"
    },
  ],
}

```

```
    {"CreatedAt": "2023-06-24T17:54:46.542000+00:00",
      "AccessGrantId": "ee8ee089-b224-4d72-85f6-975b4185a1b2",
      "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2",
      "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-9"
      },
      "Permission": "READ",
      "AccessGrantsLocationId": "12414a34-ea2e-4ddf-b425-d0d4bfcacao0",
      "GrantScope": "s3://amzn-s3-demo-bucket/prefixB*"
    },
  ],
}
```

Mit dem REST API

Sie können Amazon S3 API S3-Operationen verwenden, um die Details einer Zugriffserteilung einzusehen und alle Zugriffsberechtigungen in einer S3 Access Grants-Instance aufzulisten. Informationen zur REST API Unterstützung bei der Verwaltung von Zugriffsberechtigungen finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [GetAccessGrant](#)
- [ListAccessGrants](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details einer Zugriffsberechtigung mithilfe der AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Abrufen der Details einer Zugriffsgewährung

```
public void getAccessGrant() {
    GetAccessGrantRequest getRequest = GetAccessGrantRequest.builder()
```

```

.accountId("111122223333")
.accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE2222")
.build();
GetAccessGrantResponse getResponse = s3Control.getAccessGrant(getRequest);
LOGGER.info("GetAccessGrantResponse: " + getResponse);
}

```

Antwort:

```

GetAccessGrantResponse(
  CreatedAt=2023-06-07T05:20:26.330Z,
  AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE2222,
  AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-fd3a5086-42f7-4b34-9fad-472e2942c70e,
  Grantee=Grantee(
    GranteeType=IAM,
    GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
  ),
  Permission=READ,
  AccessGrantsLocationId=12a6710f-5af8-41f5-b035-0bc795bf1a2b,
  AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
    S3SubPrefix=prefixB*
  ),
  GrantScope=s3://amzn-s3-demo-bucket/
)

```

Example – Auflisten aller Zugriffsgewährungen in einer S3-Access-Grants-Instance

Sie können diese Parameter optional verwenden, um die Ergebnisse auf ein S3-Präfix oder eine IAM Identität zu beschränken:

- Umfang – GrantScope=s3://*bucket-name/prefix**
- Gewährungsempfänger – GranteeType=IAM und GranteeIdentifier=arn:aws:iam::*111122223333*:role/*accessGrantsConsumerRole*

```

public void listAccessGrants() {
  ListAccessGrantsRequest listRequest = ListAccessGrantsRequest.builder()
    .accountId("111122223333")
    .build();
  ListAccessGrantsResponse listResponse = s3Control.listAccessGrants(listRequest);
}

```

```
LOGGER.info("ListAccessGrantsResponse: " + listResponse);
}
```

Antwort:

```
ListAccessGrantsResponse(
  AccessGrantsList=[
    ListAccessGrantEntry(
      CreatedAt=2023-06-14T17:54:46.540z,
      AccessGrantId=dd8dd089-b224-4d82-95f6-975b4185bbaa,
      AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa,
      Grantee=Grantee(
        GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-3
      ),
      Permission=READ,
      AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcada1,
      GrantScope=s3://amzn-s3-demo-bucket/prefixA
    ),
    ListAccessGrantEntry(
      CreatedAt=2023-06-24T17:54:46.540z,
      AccessGrantId=ee8ee089-b224-4d72-85f6-975b4185a1b2,
      AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2,
      Grantee=Grantee(
        GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-9
      ),
      Permission=READ,
      AccessGrantsLocationId=12414a34-ea2e-4ddf-b425-d0d4bfcacao0,
      GrantScope=s3://amzn-s3-demo-bucket/prefixB*
    )
  ]
)
```

Löschen einer Gewährung

Sie können Zugriffsgewährungen aus Ihrer Amazon-S3-Access-Grants-Instance löschen. Sie können die Löschung einer Zugriffsgewährung nicht rückgängig machen. Nach dem Löschen einer Zugriffsberechtigung hat der Empfänger nicht länger Zugriff auf Ihre Amazon-S3-Daten.

Sie können eine Zugriffsgewährung löschen, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

So löschen Sie eine Zugriffsgewährung

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite die Registerkarte Gewährungen aus.
6. Suchen Sie die Gewährung, die Sie löschen möchten. Wenn Sie die Gewährung gefunden haben, wählen Sie das Optionsfeld neben der Gewährung aus.
7. Wählen Sie Löschen. Anschließend wird ein Dialogfeld angezeigt, in dem Sie gewarnt werden, dass diese Aktion nicht rückgängig gemacht werden kann. Wählen Sie erneut Löschen aus, um die Gewährung zu löschen.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Löschen einer Zugriffsgewährung

```
aws s3control delete-access-grant \  
--account-id 111122223333 \  
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  
  
// No response body
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung von Zugriffsberechtigungen finden Sie unter [DeleteAccessGrant](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie eine Zugriffsgewährung mithilfe der AWS SDKs. Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Löschen einer Zugriffsgewährung

```
public void deleteAccessGrant() {
    DeleteAccessGrantRequest deleteRequest = DeleteAccessGrantRequest.builder()
        .accountId("111122223333")
        .accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
        .build();
    DeleteAccessGrantResponse deleteResponse =
        s3Control.deleteAccessGrant(deleteRequest);
    LOGGER.info("DeleteAccessGrantResponse: " + deleteResponse);
}
```

Antwort:

```
DeleteAccessGrantResponse()
```

Abrufen von S3-Daten mithilfe von Zugriffsberechtigungen

Stipendiaten, denen über S3 Access Grants Zugriff auf S3-Daten gewährt wurde, müssen temporäre Anmeldeinformationen von S3 Access Grants anfordern, die sie für den Zugriff auf die S3-Daten verwenden. Weitere Informationen finden Sie unter [Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants](#). Die Empfänger verwenden dann die temporären Anmeldeinformationen, um zulässige S3-Aktionen mit den S3-Daten durchzuführen. Weitere Informationen finden Sie unter [Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden](#). Zuschussempfänger können optional eine Liste ihrer Zugangsberechtigungen für eine

anfordern AWS-Konto bevor Sie diese Anmeldeinformationen anfordern. Weitere Informationen finden Sie unter [Listen Sie die Zugriffsberechtigungen des Anrufers auf](#).

Themen

- [Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants](#)
- [Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden](#)
- [Listen Sie die Zugriffsberechtigungen des Anrufers auf](#)

Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants

Nachdem Sie Amazon S3 Access Grants verwendet haben, um [eine Zugriffsberechtigung zu erstellen](#), die AWS Identity and Access Management (IAM) Prinzipale, die Identitäten Ihres Unternehmensverzeichnisses oder autorisierte Anwendungen, die auf Ihre S3-Daten zugreifen, können Ihre Stipendiaten Anmeldeinformationen für den Zugriff auf diese Daten anfordern.

Wenn eine Anwendung oder AWS-Service verwendet den GetDataAccess API Vorgang, um S3 Access Grants im Namen eines Empfängers um Zugriff auf Ihre S3-Daten zu bitten, überprüft S3 Access Grants zunächst, ob Sie dieser Identität Zugriff auf die Daten gewährt haben. Anschließend verwendet S3 Access Grants die [AssumeRole](#) API Vorgang, um ein temporäres Anmeldeinformationstoken zu erhalten und es an den Anforderer zu senden. Dieses temporäre Anmeldeinformationstoken ist ein AWS Security Token Service (AWS STS) Token.

Die GetDataAccess-Anforderung muss den Parameter `target` enthalten, um den Umfang der S3-Daten anzugeben, für den die temporären Anmeldeinformationen gelten. Dieser `target`-Umfang kann mit dem Umfang der Gewährung oder einem Teil dieses Umfangs identisch sein. Der `target`-Umfang muss jedoch innerhalb des Umfangs der Gewährung für den Anforderer liegen. In der Anforderung muss auch der Parameter `permission` angegeben werden, der die Berechtigungsstufe für die temporären Anmeldeinformationen angibt, `READ`, `WRITE` oder `READWRITE`.

Der Anforderer kann die Berechtigungsstufe des temporären Tokens in der Anforderung für Anmeldeinformationen angeben. Mit dem Parameter `privilege` kann der Anforderer den Umfang des Zugriffs für die temporären Anmeldeinformationen innerhalb der Grenzen des Gewährungsumfangs reduzieren oder erhöhen. Der Standardwert des Parameters `privilege` ist `Default`. Das bedeutet, dass der Zielumfang der zurückgegebenen Anmeldeinformationen der ursprüngliche Umfang der Gewährung ist. Der andere mögliche Wert für `privilege` ist `Minimal`. Wenn der `target`-Umfang im Vergleich zum ursprünglichen Umfang der Gewährung reduziert wird,

wird der Umfang der temporären Anmeldeinformationen an den target-Umfang angepasst, solange der target-Umfang innerhalb des Gewährungsumfangs liegt.

Die folgende Tabelle zeigt die Auswirkungen des Parameters `privilege` auf zwei Gewährungen. Eine Gewährung hat den Umfang `S3://amzn-s3-demo-bucket1/bob/*`. Dieser Umfang umfasst das gesamte Präfix `bob/` im Bucket `amzn-s3-demo-bucket1`. Die andere Gewährung hat den Umfang `S3://amzn-s3-demo-bucket1/bob/reports/*`. Dieser Umfang umfasst nur das Präfix `bob/reports/` im Bucket `amzn-s3-demo-bucket1`.

Umfang der Gewährung	Angeforderter Umfang	Recht	Zurückgegebener Umfang	Auswirkung
<code>S3://amzn-s3-demo-bucket1/bob/*</code>	<code>amzn-s3-demo-bucket1/bob/*</code>	Default	<code>amzn-s3-demo-bucket1/bob/*</code>	Der Anforderer hat Zugriff auf alle Objekte im Bucket <code>amzn-s3-demo-bucket1</code> , deren Schlüsselnamen mit dem Präfix <code>bob/</code> beginnen.
<code>S3://amzn-s3-demo-bucket1/bob/*</code>	<code>amzn-s3-demo-bucket1/bob/</code>	Minimal	<code>amzn-s3-demo-bucket1/bob/</code>	Ohne das Platzhaltzeichen (*) nach dem Präfixnamen <code>bob/</code> hat der Anforderer nur Zugriff auf das Objekt mit dem Namen <code>bob/</code> im Bucket <code>amzn-s3-demo-bucket1</code> . Ein solches Objekt ist nicht üblich. Der Anforderer hat keinen Zugriff auf andere Objekte, auch nicht auf Objekte, deren Schlüsselnamen mit dem Präfix <code>bob/</code> beginnen.
<code>S3://amzn-s3-demo-bucket1/bob/images/*</code>	<code>amzn-s3-demo-bucket1/bob/</code>	Minimal	<code>amzn-s3-demo-bucket1/bob/images/*</code>	Der Anforderer hat Zugriff auf alle Objekte

Umfang der Gewährung	Angeforderter Umfang	Recht	Zurückgegebener Umfang	Auswirkung
<i>emo-bucke</i> <i>t1 /bob/</i> <i>*</i>	<i>bucke</i> <i>t1 /bob/</i> <i>images/</i> <i>*</i>			im Bucket <i>amzn-s3-demo-bucket1</i> , deren Schlüsselnamen mit dem Präfix <i>bob/images/</i> beginnen.
<i>S3://amzn-s3-demo-bucke</i> <i>t1 /bob/</i> <i>repo</i> <i>rts/</i> <i>rts/*</i>	<i>amzn-s3-demo-bucke</i> <i>t1 /bob/</i> <i>repo</i> <i>rts/</i> <i>file.</i> <i>txt</i>	Default	<i>amzn-s3-demo-bucke</i> <i>t1 /bob/reports/*</i>	Der Anforderer hat Zugriff auf alle Objekte im Bucket <i>amzn-s3-demo-bucket1</i> , deren Schlüsselnamen mit dem Präfix <i>bob/reports</i> beginnen. Dies entspricht dem Umfang der entsprechenden Gewährung.
<i>S3://amzn-s3-demo-bucke</i> <i>t1 /bob/</i> <i>repo</i> <i>rts/</i> <i>rts/*</i>	<i>amzn-s3-demo-bucke</i> <i>t1 /bob/</i> <i>repo</i> <i>rts/</i> <i>file.</i> <i>txt</i>	Minimal	<i>amzn-s3-demo-bucke</i> <i>t1 /bob/reports/</i> <i>file.txt</i>	Der Anforderer hat nur Zugriff auf das Objekt mit dem Schlüsselnamen <i>bob/reports/file.txt</i> im Bucket <i>amzn-s3-demo-bucke</i> <i>t1</i> . Der Anforderer hat keinen Zugriff auf andere Objekte.

Der Parameter `durationSeconds` legt die Gültigkeitsdauer der temporären Anmeldeinformationen in Sekunden fest. Der Standardwert ist 3600 Sekunden (1 Stunde). Der Anforderer (Empfänger) kann jedoch einen Bereich von 900 Sekunden (15 Minuten) bis 43200 Sekunden (12 Stunden) angeben. Wenn der Empfänger einen höheren Wert als diesen Höchstwert anfordert, schlägt die Anforderung fehl.

Note

Wenn der Speicherort ein Objekt ist, legen Sie in Ihrer Anforderung eines temporären Tokens den Wert des Parameters `targetType` auf `Object` fest. Dieser Parameter ist nur erforderlich, wenn es sich der Speicherort ein Objekt ist und die Berechtigungsstufe `Minimal` ist. Wenn der Speicherort ein Bucket oder ein Präfix ist, müssen Sie diesen Parameter nicht angeben.

Weitere Informationen finden Sie [GetDataAccess](#) in der Amazon Simple Storage Service API Reference.

Sie können temporäre Anmeldeinformationen anfordern, indem Sie AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example Anfordern temporärer Anmeldeinformationen

Anfrage:

```
aws s3control get-data-access \  
--account-id 111122223333 \  
--target s3://amzn-s3-demo-bucket/prefixA* \  
--permission READ \  
--privilege Default \  
--region us-east-2
```

Antwort:

```
{  
  "Credentials": {  
    "AccessKeyId": "Example-key-id",
```

```
"SecretAccessKey": "Example-access-key",
"SessionToken": "Example-session-token",
"Expiration": "2023-06-14T18:56:45+00:00"},
  "MatchedGrantTarget": "s3://amzn-s3-demo-bucket/prefixA**"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Anforderung temporärer Anmeldeinformationen von S3 Access Grants finden Sie [GetDataAccess](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Dieser Abschnitt enthält ein Beispiel dafür, wie Stipendiaten temporäre Anmeldeinformationen von S3 Access Grants anfordern, indem sie den AWS SDKs.

Java

Das folgende Codebeispiel gibt die temporären Anmeldeinformationen zurück, die der Gewährungsempfänger für den Zugriff auf Ihre S3-Daten verwendet. Wenn Sie dieses Codebeispiel verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Daten.

Example Abrufen temporärer Anmeldeinformationen

Anfrage:

```
public void getDataAccess() {
    GetDataAccessRequest getDataAccessRequest = GetDataAccessRequest.builder()
        .accountId("111122223333")
        .permission(Permission.READ)
        .privilege(Privilege.MINIMAL)
        .target("s3://amzn-s3-demo-bucket/prefixA*")
        .build();
    GetDataAccessResponse getDataAccessResponse =
        s3Control.getDataAccess(getDataAccessRequest);
    LOGGER.info("GetDataAccessResponse: " + getDataAccessResponse);
}
```

Antwort:

```
GetDataAccessResponse(  
  Credentials=Credentials(  
    AccessKeyId="Example-access-key-id",  
    SecretAccessKey="Example-secret-access-key",  
    SessionToken="Example-session-token",  
    Expiration=2023-06-07T06:55:24Z  
  )  
)
```

Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden

Nachdem ein Stipendiat im Rahmen seiner [Zugriffsgewährung temporäre Anmeldeinformationen](#) erhalten hat, kann er diese temporären Anmeldeinformationen verwenden, um Amazon S3 API S3-Operationen für den Zugriff auf Ihre Daten aufzurufen.

Zuschussempfänger können auf S3-Daten zugreifen, indem sie AWS Command Line Interface (AWS CLI), das AWS SDKs und der Amazon S3 RESTAPI. Zusätzlich können Sie die verwenden [AWS Python](#) - und [Java-Plugins](#) zum Aufrufen von S3 Access Grants

Verwendung der AWS CLI

Nachdem der Gewährungsempfänger seine temporären Anmeldeinformationen von S3 Access Grants erhalten hat, kann er mit diesen Anmeldeinformationen ein Profil einrichten, um die Daten abzurufen.

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example – Einrichten eines Profils

```
aws configure set aws_access_key_id "$accessKey" --profile access-grants-consumer-access-profile  
aws configure set aws_secret_access_key "$secretKey" --profile access-grants-consumer-access-profile  
aws configure set aws_session_token "$sessionToken" --profile access-grants-consumer-access-profile
```

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Abrufen der S3-Daten

Der Stipendiat kann das verwenden [get-object](#) AWS CLI Befehl, um auf die Daten zuzugreifen. Der Stipendiat kann auch Folgendes verwenden [put-object](#), [ls](#), und andere S3 AWS CLI Befehle.

```
aws s3api get-object \  
--bucket amzn-s3-demo-bucket1 \  
--key myprefix \  
--region us-east-2 \  
--profile access-grants-consumer-access-profile
```

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Stipendiaten mithilfe der AWS SDKs.

Java

Beispiele dafür, wie Sie S3-Daten mithilfe temporärer Anmeldeinformationen abrufen können, finden Sie unter So [rufen Sie ein Objekt mit dem AWS SDKs](#) und [Amazon S3 S3-Codebeispiele für AWS SDK for Java 2.x](#).

Unterstützte S3-Aktionen in S3 Access Grants

Ein Empfänger kann die von S3 Access Grants bereitgestellten temporären Anmeldeinformationen verwenden, um S3-Aktionen mit den S3-Daten durchzuführen, auf die er Zugriff hat. Im Folgenden finden Sie eine Liste der zulässigen S3-Aktionen, die ein Empfänger ausführen kann. Welche Aktionen zulässig sind, hängt von der in der Zugriffsgewährung gewährten Berechtigungsstufe ab, entweder READ, WRITE oder READWRITE

Note

Zusätzlich zu den unten aufgeführten Amazon S3-Berechtigungen kann Amazon S3 die AWS Key Management Service (AWS KMS) [Decrypt](#) (kms:decrypt) READ -Berechtigung oder die AWS KMS [GenerateDataKey](#) (kms:generateDataKey) WRITE Erlaubnis. Diese Berechtigungen ermöglichen keinen direkten Zugriff auf AWS KMS Schlüssel.

IAMS3-Aktion	APIAktion und Dokument	S3 Access erteilt die Erlaubnis	S3-Ressource			
s3:GetObject	GetObject	READ	Object			
s3:GetObjectVersion	GetObject	READ	Object			
s3:GetObjectAcl	GetObjectAcl	READ	Object			
s3:GetObjectVersionAcl	GetObjectAcl	READ	Object			
s3:ListMultipartUploads	ListParts	READ	Object			
s3:PutObject	PutObject , CreateMultipartUpload , UploadPart , UploadPartCopy , CompleteMultipartUpload	WRITE	Object			
s3:PutObjectAcl	PutObjectAcl	WRITE	Object			
s3:PutObjectVersionAcl	PutObjectAcl	WRITE	Object			
s3:DeleteObject	DeleteObject	WRITE	Object			

IAMS3-Aktion	APIAktion und Dokument	S3 Access erteilt die Erlaubnis	S3-Ressource			
s3:DeleteObjectVersion	DeleteObject	WRITE	Object			
s3:AbortMultipartUpload	AbortMultipartUpload	WRITE	Object			
s3:ListBucket	HeadBucket , ListObjectsV2 , ListObjects	READ	Bucket			
s3:ListBucketVersions	ListObjectVersions	READ	Bucket			
s3:ListBucketMultipartUploads	ListMultipartUploads	READ	Bucket			

Listen Sie die Zugriffsberechtigungen des Anrufers auf

S3-Datenbesitzer können S3 Access Grants verwenden, um Zugriffsberechtigungen für zu erstellen AWS Identity and Access Management (IAM) Identitäten oder für AWS IAM Identity Center Identitäten im Unternehmensverzeichnis. IAM-Identitäten und IAM Identity Center-Verzeichnisidentitäten können wiederum die verwenden, `ListCallerAccessGrants` API um alle Amazon S3 S3-Buckets, -Präfixe und Objekte aufzulisten, auf die sie zugreifen können, wie in ihren S3 Access Grants definiert. Verwenden Sie diese OptionAPI, um alle S3-Daten zu ermitteln, auf die eine IAM oder eine Verzeichnisidentität über S3 Access Grants zugreifen kann.

Sie können diese Funktion verwenden, um Anwendungen zu erstellen, die die Daten anzeigen, auf die bestimmte Endbenutzer zugreifen können. Zum Beispiel die AWS Storage Browser for S3, eine Open-Source-UI-Komponente, die Kunden für den Zugriff auf S3-Buckets verwenden, verwendet

diese Funktion, um Endbenutzern die Daten zu präsentieren, auf die sie in Amazon S3 Zugriff haben, basierend auf ihren S3 Access Grants. Ein anderes Beispiel: Wenn Sie eine Anwendung zum Durchsuchen, Hochladen oder Herunterladen von Daten in Amazon S3 erstellen, können Sie diese Funktion verwenden, um eine Baumstruktur in Ihrer Anwendung zu erstellen, die ein Endbenutzer dann durchsuchen kann.

Der Empfänger, unabhängig davon, ob es sich um eine IAM Identität oder eine Unternehmensverzeichnisidentität handelt, kann eine Liste seiner Zugriffsberechtigungen abrufen, indem er die AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example Listet die Zugriffsberechtigungen eines Anrufers auf

Anfrage:

```
aws s3control list-caller-access-grants \  
--account-id 111122223333 \  
--region us-east-2 \  
--max-results 5
```

Antwort:

```
{  
  "NextToken": "6J9S...",  
  "CallerAccessGrantsList": [  
    {  
      "Permission": "READWRITE",  
      "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix1/*",  
      "ApplicationArn": "NA"  
    },  
    {  
      "Permission": "READWRITE",  
      "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix2/*",
```

```

  "ApplicationArn": "ALL"
},
{
  "Permission": "READWRITE",
  "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix3/*",
  "ApplicationArn": "arn:aws:sso::111122223333:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
}
]
}

```

Example Listet die Zugriffsberechtigungen eines Anrufers für einen Bucket auf

Sie können den Umfang der Ergebnisse mithilfe des `grantscope` Parameters einschränken.

Anfrage:

```

aws s3control list-caller-access-grants \
--account-id 111122223333 \
--region us-east-2
--grant-scope "s3://amzn-s3-demo-bucket*"
--max-results 1000

```

Antwort:

```

{
  "NextToken": "6J9S...",
  "CallerAccessGrantsList": [
    {
      "Permission": "READ",
      "GrantScope": "s3://amzn-s3-demo-bucket*",
      "ApplicationArn": "ALL"
    },
    {
      "Permission": "READ",
      "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/*",
      "ApplicationArn": "arn:aws:sso::111122223333:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
    }
  ]
}

```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für das Abrufen einer Liste der Zugriffsberechtigungen des API Anrufers finden Sie [ListCallerAccessGrants](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Dieser Abschnitt enthält ein Beispiel dafür, wie Stipendiaten temporäre Anmeldeinformationen von S3 Access Grants anfordern, indem sie den AWS SDKs.

Java

Das folgende Codebeispiel gibt die Zugriffsberechtigungen des API Anrufers auf die S3-Daten einer bestimmten Person zurück AWS-Konto. Um dieses Codebeispiel zu verwenden, ersetzen Sie das *user input placeholders* durch Ihre eigenen Informationen.

Example Listet die Zugriffsberechtigungen eines Anrufers auf

Anfrage:

```
Public void ListCallerAccessGrants() {
    ListCallerAccessGrantsRequest listRequest = ListCallerAccessGrantsRequest.builder()
        .withMaxResults(1000)
        .withGrantScope("s3://")
        .accountId("111122223333");
    ListCallerAccessGrantsResponse listResponse =
        s3control.listCallerAccessGrants(listRequest);
    LOGGER.info("ListCallerAccessGrantsResponse: " + listResponse);
}
```

Antwort:

```
ListCallerAccessGrantsResponse(
    CallerAccessGrantsList=[
    ListCallerAccessGrantsEntry(
        S3Prefix=s3://amzn-s3-demo-bucket/prefix1/,
        Permission=READ,
        ApplicationArn=ALL
    )
    ])
```

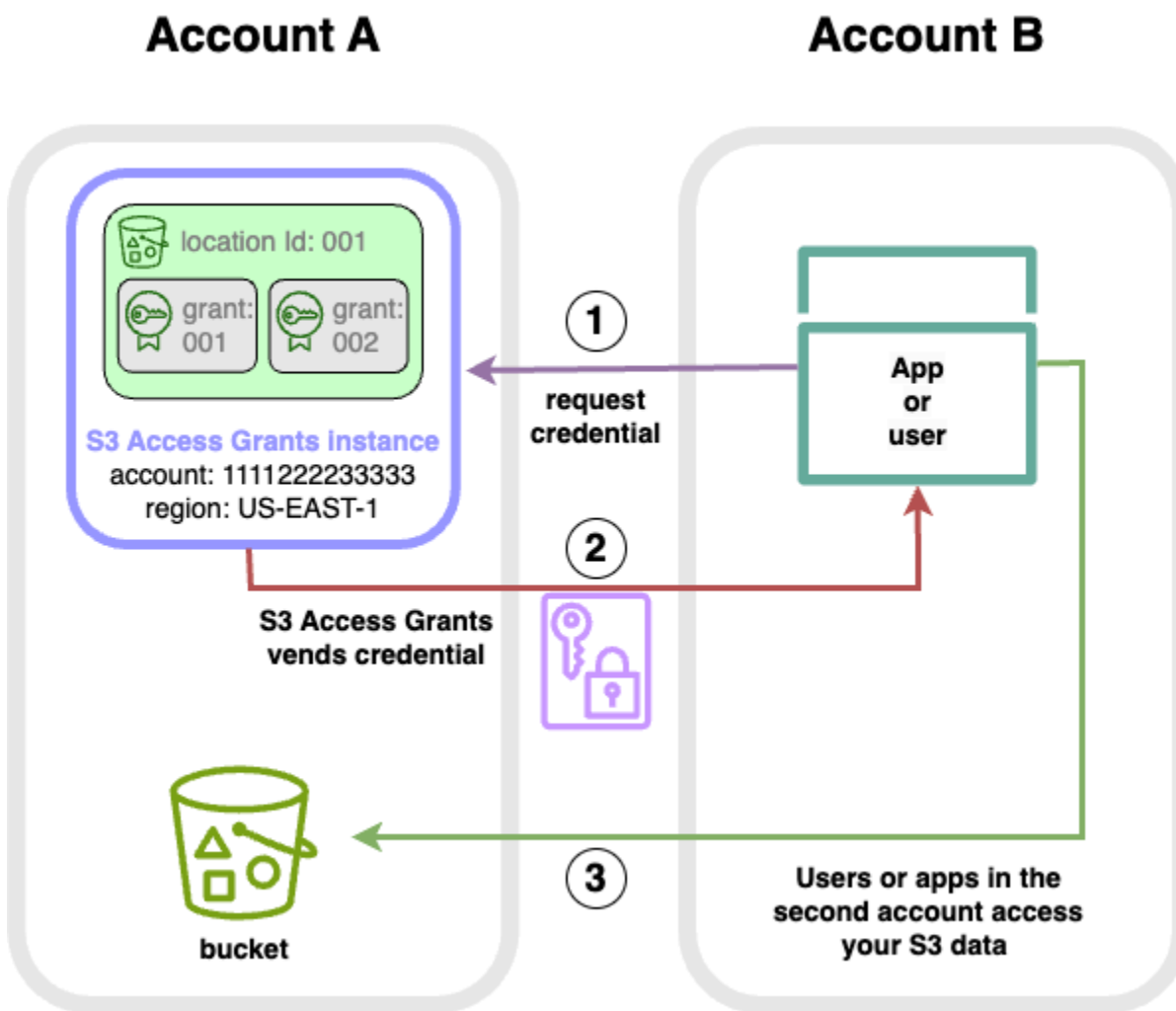
Kontoübergreifender Zugriff mit S3 Access Grants

Mit S3 Access Grants können Sie Amazon S3 S3-Datenzugriff auf Folgendes gewähren:

- AWS Identity and Access Management (IAM) Identitäten innerhalb Ihres Kontos
- IAM Identitäten in anderen AWS Konten
- Verzeichnisbenutzer oder -gruppen in Ihrem AWS IAM Identity Center Instance

Konfigurieren Sie zunächst den kontoübergreifenden Zugriff für das andere Konto. Dazu gehört die Gewährung des Zugriffs auf Ihre S3 Access Grants-Instanz mithilfe einer Ressourcenrichtlinie. Gewähren Sie anschließend mithilfe von Zuschüssen Zugriff auf Ihre S3-Daten (Buckets, Präfixe oder Objekte).

Nachdem Sie den kontoübergreifenden Zugriff konfiguriert haben, kann das andere Konto temporäre Zugangsdaten für Ihre Amazon S3 S3-Daten von S3 Access Grants anfordern. Die folgende Abbildung zeigt den Benutzerablauf für den kontoübergreifenden S3-Zugriff über S3 Access Grants:



1. Benutzer oder Anwendungen in einem zweiten Konto (B) fordern Anmeldeinformationen von der S3 Access Grants-Instance in Ihrem Konto (A) an, in der die Amazon S3 S3-Daten gespeichert sind. Weitere Informationen finden Sie unter [Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants](#).
2. Die S3 Access Grants-Instance in Ihrem Konto (A) gibt temporäre Anmeldeinformationen zurück, wenn es eine Zuweisung gibt, die dem zweiten Konto Zugriff auf Ihre Amazon S3 S3-Daten gewährt. Weitere Informationen zu Zugriffsberechtigungen finden Sie unter

Eine individuelle Zugriffsberechtigung in einer S3 Access Grants-Instance ermöglicht eine bestimmte Identität — eine AWS Identity and Access Management (IAM) Principal oder ein Benutzer oder eine Gruppe in einem Unternehmensverzeichnis — um Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist. Ein Standort ordnet Buckets oder Präfixe einer Rolle zu. IAM S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an Empfänger zu verkaufen.

Nachdem Sie mindestens einen Standort in Ihrer S3 Access Grants-Instanz registriert haben, können Sie eine Zugriffsberechtigung erstellen.

Der Empfänger kann ein IAM Benutzer oder eine Rolle oder ein Verzeichnisbenutzer oder eine Gruppe sein. Ein Verzeichnisbenutzer ist ein Benutzer aus Ihrem Unternehmensverzeichnis oder einer externen Identitätsquelle, den Sie mit Ihrer S3 Access Grants-Instanz verknüpft haben. Weitere Informationen finden Sie unter S3 Access Grants und Unternehmensverzeichnisidentitäten. Um einen Zuschuss für einen bestimmten Verzeichnisbenutzer oder eine bestimmte Verzeichnisgruppe aus IAM Identity Center zu erstellen, suchen Sie in IAM Identity Center nach dem GUID, anhand dessen IAM Identity Center diesen Benutzer identifiziert, a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 z. B. Weitere Informationen zur Verwendung von IAM Identity Center zum Anzeigen von Benutzerinformationen finden Sie unter Benutzer- und Gruppenzuweisungen anzeigen in AWS IAM Identity Center Benutzerleitfaden.

Sie können Zugriff auf einen Bucket, ein Präfix oder ein Objekt gewähren. Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, mit der die Objekte in einem Bucket organisiert werden. Dabei kann es sich um eine beliebige Zeichenfolge mit zulässigen Zeichen handeln, z. B. Objektschlüsselnamen in Ihrem Bucket, die mit dem `engineering/` Präfix beginnen.

Themen

- Erstellen von Gewährungen
 - Anzeigen einer Gewährung
 - Löschen einer Gewährung
-

Erstellen von Gewährungen

Eine individuelle Zugriffsgewährung in einer S3 Access Grants-Instanz ermöglicht eine bestimmte Identität — eine AWS Identity and Access Management (IAM) Principal oder ein Benutzer oder eine Gruppe in einem Unternehmensverzeichnis — um Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist. Ein Standort ordnet Buckets oder Präfixe einer Rolle zu. IAM S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an Empfänger zu verkaufen.

Nachdem Sie mindestens einen Standort in Ihrer S3 Access Grants-Instanz registriert haben, können Sie eine Zugriffsberechtigung erstellen.

Der Empfänger kann ein IAM Benutzer oder eine Rolle oder ein Verzeichnisbenutzer oder eine Gruppe sein. Ein Verzeichnisbenutzer ist ein Benutzer aus Ihrem Unternehmensverzeichnis oder einer externen Identitätsquelle, den Sie mit Ihrer S3 Access Grants-Instanz verknüpft haben. Weitere Informationen finden Sie unter [S3 Access Grants und Unternehmensverzeichnisidentitäten](#). Um einen Zuschuss für einen bestimmten Verzeichnisbenutzer oder eine bestimmte Verzeichnisgruppe aus IAM Identity Center zu erstellen, suchen Sie in IAM Identity Center nach dem GUID, anhand dessen IAM Identity Center diesen Benutzer identifiziert, a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 z. B. Weitere Informationen zur Verwendung von IAM Identity Center zum Anzeigen von Benutzerinformationen finden Sie unter [Benutzer- und Gruppenzuweisungen anzeigen in AWS IAM Identity Center Benutzerleitfaden](#).

Sie können Zugriff auf einen Bucket, ein Präfix oder ein Objekt gewähren. Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, mit der die Objekte in einem Bucket organisiert werden. Dabei kann es sich um eine beliebige Zeichenfolge mit zulässigen Zeichen handeln, z. B. Objektschlüsselnamen in Ihrem Bucket, die mit dem `engineering/` Präfix beginnen.

Unterpräfix

Wenn Sie Zugriff auf einen registrierten Standort gewähren, können Sie das Subprefix Feld verwenden, um den Zugriffsbereich auf eine Teilmenge des Standortbereichs einzuschränken. Wenn es sich bei dem registrierten Standort, den Sie für die Gewährung auswählen, um den Standard-S3-Pfad (`s3://`) handelt, müssen Sie den Geltungsbereich der Gewährung einschränken. Sie können keine Zugriffsberechtigung für den Standardspeicherort (`s3://`) erstellen, wodurch der Empfänger Zugriff auf jeden Bucket in einem AWS-Region. Stattdessen müssen Sie den Umfang des Zuschusses auf einen der folgenden Bereiche einschränken:

- Ein Eimer: `s3://bucket/*`
- Ein Präfix innerhalb eines Buckets: `s3://bucket/prefix*`
- Ein Präfix innerhalb eines Präfixes: `s3://bucket/prefixA/prefixB*`
- Ein Objekt: `s3://bucket/object-key-name`

Wenn Sie eine Zugriffsberechtigung erstellen, bei der es sich bei dem registrierten Standort um einen Bucket handelt, können Sie eine der folgenden Angaben in das Subprefix Feld eingeben, um den Umfang der Gewährung einzuschränken:

- Ein Präfix innerhalb des Buckets: *prefix**

- Ein Präfix innerhalb eines Präfixes: *prefixA/prefixB**

- Ein Objekt: */object-key-name*

Nachdem Sie den Zuschuss erstellt haben, wird der Umfang der Gewährung, der in der Amazon S3 S3-Konsole angezeigt wird GrantScope, API oder AWS Command Line Interface (AWS CLI) Die Antwort ist das Ergebnis der Verkettung des Standortpfads mit dem Subprefix. Dieser verkettete Pfad muss mit dem S3-Bucket, Präfix oder Objekt übereinstimmen, dem Sie Zugriff gewähren möchten.

Note

- Wenn Sie eine Zugriffserteilung erstellen müssen, die nur Zugriff auf ein Objekt gewährt, müssen Sie angeben, dass der Gewährungstyp für ein Objekt gilt.

Um dies in einem API Aufruf oder einem CLI Befehl zu tun, übergeben Sie den

s3PrefixType Parameter mit dem Wert object. Wenn Sie in der Amazon S3

S3-Konsole den Grant erstellen, nachdem Sie einen Standort ausgewählt haben, aktivieren Sie unter Grant Scope das Kontrollkästchen Grant scope is an object.

- Sie können keine Gewährung für einen Bucket erstellen, der noch nicht vorhanden ist. Sie können jedoch einen Zuschuss für ein Präfix erstellen, das noch nicht existiert.

- Die maximale Anzahl von Zuschüssen, die Sie in Ihrer S3 Access Grants-Instanz erstellen können, finden Sie unter Limits für S3 Access Grants.

Sie können mithilfe der Amazon S3 S3-Konsole eine Zugriffsgewährung erstellen. AWS CLI, der Amazon S3 REST API und AWS SDKs.

Verwenden der S3-Konsole

So erstellen Sie eine Zugriffsgewährung

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.

3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

Wenn Sie die S3-Access-Grants-Instance zum ersten Mal verwenden, müssen Sie [Schritt 2 – Registrieren eines Speicherorts](#) abgeschlossen haben und zu Schritt 3 des Assistenten Access-Grants-Instance einrichten gewechselt sein. Wenn Sie bereits über eine S3-Access-Grants-Instance verfügen, wählen Sie Details anzeigen und dann auf der Registerkarte Gewährungen die Option Gewährung erstellen aus.

- a. Wählen Sie im Abschnitt Gewährungsumfang einen registrierten Standort aus oder geben Sie diesen ein.

Wenn Sie den `s3://`-Standardspeicherort ausgewählt haben, können Sie im Feld Unterpräfix den Umfang der Zugriffsgewährung einschränken. Weitere Informationen finden Sie unter [Unterpräfix](#). Wenn Sie lediglich einem Objekt Zugriff gewähren, wählen Sie Gewährungsumfang ist ein Objekt aus.

- b. Wählen Sie unter Berechtigungen und Zugriff die Stufe der Berechtigung aus, Lesen, Schreiben oder beides.

Wählen Sie dann den Empfängertyp aus. Wenn Sie Ihr Unternehmensverzeichnis zu IAM Identity Center hinzugefügt und diese IAM Identity Center-Instance mit Ihrer S3 Access Grants-Instance verknüpft haben, können Sie die Verzeichnisidentität aus IAM Identity Center auswählen. Wenn Sie diese Option wählen, rufen Sie die ID des Benutzers oder der Gruppe aus IAM Identity Center ab und geben Sie sie in diesen Abschnitt ein.

Wenn der Typ des Empfängers ein IAM Benutzer oder eine Rolle ist, wählen Sie IAMPrincipal. Wählen Sie unter IAMPrinzipaltyp die Option Benutzer oder Rolle aus.

Wählen Sie dann unter IAMHauptbenutzer entweder aus der Liste aus, oder geben Sie die ID der Identität ein.

- c. Um die S3-Access-Grants-Gewährung zu erstellen, wählen Sie Weiter oder Gewährung erstellen aus.

4. Wenn Weiter oder Gewährung erstellen deaktiviert ist:

Gewährung kann nicht erstellt werden

- Möglicherweise müssen Sie in Ihrer S3-Access-Grants-Instance zuerst [einen Speicherort registrieren](#).

- Möglicherweise besitzen Sie die Berechtigung `s3:CreateAccessGrant` nicht, die für die Erstellung Zugriffsgewährung erforderlich ist. Nehmen Sie Kontakt mit Ihrem Kontoadministrator auf.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI in der AWS Command Line Interface Benutzerleitfaden](#).

Die folgenden Beispiele zeigen, wie Sie eine Zugriffserteilungsanforderung für einen IAM Prinzipal und eine Zugriffserteilungsanforderung für einen Benutzer oder eine Gruppe im Unternehmensverzeichnis erstellen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Note

Wenn Sie eine Zugriffsgewährung erstellen, die Zugriff auf ein einzelnes Objekt gewährt, geben Sie den erforderlichen Parameter `--s3-prefix-type Object` an.

Example Erstellen Sie eine Anfrage zur Zugriffsgewährung für einen IAM Principal

```
aws s3control create-access-grant \
```

```
--account-id 111122223333 \
```

```
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
```

```
--access-grants-location-configuration S3SubPrefix=prefixB* \
```

```
--permission READ \
```

```
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::123456789012:user/data-
```

```
consumer-3
```

Example Erstellen einer Zugriffsgewährungsantwort

```
{"CreatedAt": "2023-05-31T18:41:34.663000+00:00",
```

```
"AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
"AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
```

```
"GranteeType": "IAM",  
"GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"  
},  
"AccessGrantsLocationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
"AccessGrantsLocationConfiguration": {  
  "S3SubPrefix": "prefixB*"  
},  
"GrantScope": "s3://DOC-BUCKET-EXAMPLE/prefix*"  
"Permission": "READ"  
}
```

Erstellen einer Anforderung für eine Zugriffsgewährung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe

Um eine Zugriffserteilungsanforderung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe zu erstellen, müssen Sie zuerst die GUID für den Verzeichnisbenutzer oder die Verzeichnisgruppe abrufen, indem Sie einen der folgenden Befehle ausführen.

Example Rufen Sie eine GUID für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe ab

Sie finden den Namen GUID eines IAM Identity Center-Benutzers über die IAM Identity Center-Konsole oder mithilfe der AWS CLI or AWS SDKs. Der folgende Befehl listet die Benutzer in der angegebenen IAM Identity Center-Instanz mit ihren Namen und Kennungen auf.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Dieser Befehl listet die Gruppen in der angegebenen IAM Identity Center-Instanz auf.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Erstellen einer Zugriffsgewährung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe

Dieser Befehl ähnelt der Erstellung eines Zuschusses für IAM Benutzer oder Rollen, mit der Ausnahme, dass der Typ des Empfängers DIRECTORY_USER oder DIRECTORY_GROUP ist und die Empfänger-ID die GUID für den Verzeichnisbenutzer oder die Gruppe ist.

```
--access-grants-location-configuration S3SubPrefix="amzn-s3-demo-bucket/rafael/*"
```

```
\
```

```
--permission READWRITE \
```

```
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier=83d43802-00b1-7054-db02-
```

```
f1d683aacba5 \
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung von Zugriffsberechtigungen finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [CreateAccessGrant](#)
- [DeleteAccessGrant](#)
- [GetAccessGrant](#)
- [ListAccessGrants](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie eine Zugriffsberechtigung mithilfe der AWS SDKs.

Java

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Daten.

Note

Wenn Sie eine Zugriffsgewährung erstellen, die Zugriff auf ein einzelnes Objekt gewährt, geben Sie den erforderlichen Parameter `.s3PrefixType(S3PrefixType.Object)` an.

Example Erstellen einer Anforderung für eine Zugriffsgewährung

```
public void createAccessGrant() {
```

```
    CreateAccessGrantRequest createRequest = CreateAccessGrantRequest.builder()
```

```
        accountId("111122223333")
```

```
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa")
```

```
.permission("READ")
```

```
.accessGrantsLocationConfiguration(AccessGrantsLocationConfiguration.builder().s3SubPre
```

```
.grantee(Grantee.builder().granteeType("IAM").granteeIdentifier("arn:aws:iam::111122223
```

```
data-consumer-3").build())
```

```
.build();
```

```
CreateAccessGrantResponse createResponse =
```

```
s3Control.createAccessGrant(createRequest);
```

```
LOGGER.info("CreateAccessGrantResponse: " + createResponse);
```

```
}
```

Example Erstellen einer Zugriffsgewährungsantwort

```
CreateAccessGrantResponse(
```

```
CreatedAt=2023-06-07T05:20:26.330Z,
```

```
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
```

```
AccessGrantArn=arn:aws:s3:us-east-2:444455556666:access-grants/default/grant/
```

```
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
```

```
Grantee=Grantee(
```

```
GranteeType=IAM,
```

```
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
```

```
),
```

```
AccessGrantsLocationId=a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa,
```

```
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
```

```
S3SubPrefix=prefixB*
```

Verwenden der S3-Konsole

So zeigen Sie die Details einer Zugriffsgewährung an

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite die Registerkarte Gewährungen aus.
6. Suchen Sie im Bereich Gewährungen nach der Zugriffsgewährung, die Sie anzeigen möchten. Sie können die Liste der Gewährungen über das Suchfeld filtern.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example – Abrufen der Details einer Zugriffsgewährung

```
aws s3control get-access-grant \  
--account-id 111122223333 \  
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Antwort:

```
{  
  "CreatedAt": "2023-05-31T18:41:34.663000+00:00",  
  "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/  
grant-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
"Grantee": {
```

```

    },
    "Permission": "READ",
    "AccessGrantsLocationId": "12a6710f-5af8-41f5-b035-0bc795bf1a2b",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://amzn-s3-demo-bucket/"
}

```

Example – Auflisten aller Zugriffsgewährungen in einer S3-Access-Grants-Instance

Sie können optional die folgenden Parameter verwenden, um die Ergebnisse auf ein S3-Präfix zu beschränken oder AWS Identity and Access Management (IAM) Identität:

- Unterpräfix – `--grant-scope s3://bucket-name/prefix*`
- IAM Identität — `--grantee-type IAM` und `--grantee-identifizier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```

aws s3control list-access-grants \
--account-id 111122223333

```

Antwort:

```

{
  "AccessGrantsList": [{"CreatedAt": "2023-06-14T17:54:46.542000+00:00",
    "AccessGrantId": "dd8dd089-b224-4d82-95f6-975b4185bbaa",
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/grant/dd8dd089-b224-4d82-95f6-975b4185bbaa",

```

```

    "Grantee": {

```

```

      "GranteeType": "IAM",

```

```

      "GranteeIdentifier": "arn:aws:iam::111122223333:user/dave"
    }
  }
}

```



```
"Grantee": {  
  "GranteeType": "IAM",  
  "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-  
consumer-9",  
},  
  "Permission": "READ",  
  "AccessGrantsLocationId": "12414a34-ea2e-4ddf-b425-d0d4bfcacao0",  
  "GrantScope": "s3://amzn-s3-demo-bucket/prefixB*"  
},  
]  
}
```

Mit dem REST API

Sie können Amazon S3 API S3-Operationen verwenden, um die Details einer Zugriffserteilung einzusehen und alle Zugriffsberechtigungen in einer S3 Access Grants-Instance aufzulisten. Informationen zur REST API Unterstützung bei der Verwaltung von Zugriffsberechtigungen finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [GetAccessGrant](#)

- [ListAccessGrants](#)

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details einer Zugriffsberechtigung mithilfe der AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

```

GetAccessGrantResponse getResponse = s3Control.getAccessGrant(getRequest);

LOGGER.info("GetAccessGrantResponse: " + getResponse);

}

```

Antwort:

```

GetAccessGrantResponse(
  CreatedAt=2023-06-07T05:20:26.330Z,
  AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE22222,
  AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-fd3a5086-42f7-4b34-9fad-472e2942c70e,
  Grantee=Grantee(
    GranteeType=IAM,
    GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
  ),
  Permission=READ,
  AccessGrantsLocationId=12a6710f-5af8-41f5-b035-0bc795bf1a2b,
  AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
    S3SubPrefix=prefixB*
  ),
  GrantScope=s3://amzn-s3-demo-bucket/
)

```

Example – Auflisten aller Zugriffsgewährungen in einer S3-Access-Grants-Instance

Sie können diese Parameter optional verwenden, um die Ergebnisse auf ein S3-Präfix oder eine IAM Identität zu beschränken:

Antwort:

```
ListAccessGrantsResponse(  
  AccessGrantsList=[  
    ListAccessGrantEntry(  
      CreatedAt=2023-06-14T17:54:46.540z,  
      AccessGrantId=dd8dd089-b224-4d82-95f6-975b4185bbaa,  
      AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/  
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa,  
      Grantee=Grantee(  
        GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-  
consumer-3  
      ),  
      Permission=READ,  
      AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcada1,  
      GrantScope=s3://amzn-s3-demo-bucket/prefixA  
    ),  
    ListAccessGrantEntry(  
      CreatedAt=2023-06-24T17:54:46.540z,  
      AccessGrantId=ee8ee089-b224-4d72-85f6-975b4185a1b2,  
      AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/  
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2,  
      Grantee=Grantee(  
        GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-  
consumer-3  
      ),  
      Permission=READ,  
      AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcada1,  
      GrantScope=s3://amzn-s3-demo-bucket/prefixA  
    )  
  ]  
)
```

Sie können eine Zugriffsgewährung löschen, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der AWS SDKs.

Verwenden der S3-Konsole

So löschen Sie eine Zugriffsgewährung

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite die Registerkarte Gewährungen aus.
6. Suchen Sie die Gewährung, die Sie löschen möchten. Wenn Sie die Gewährung gefunden haben, wählen Sie das Optionsfeld neben der Gewährung aus.
7. Wählen Sie Löschen. Anschließend wird ein Dialogfeld angezeigt, in dem Sie gewarnt werden, dass diese Aktion nicht rückgängig gemacht werden kann. Wählen Sie erneut Löschen aus, um die Gewährung zu löschen.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Löschen einer Zugriffsgewährung

```
aws s3control delete-access-grant \
```

```
--account-id 111122223333 \
```

```
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

```
// No response body
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung von Zugriffsberechtigungen finden Sie unter [DeleteAccessGrant](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie eine Zugriffsgewährung mithilfe der AWS SDKs. Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Löschen einer Zugriffsgewährung

```
public void deleteAccessGrant() {  
    DeleteAccessGrantRequest deleteRequest = DeleteAccessGrantRequest.builder()  
        .accountId("111122223333")  
        .accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")  
        .build();  
    DeleteAccessGrantResponse deleteResponse =  
        s3Control.deleteAccessGrant(deleteRequest);  
    LOGGER.info("DeleteAccessGrantResponse: " + deleteResponse);  
}
```

Antwort:

```
DeleteAccessGrantResponse()
```

- Benutzer oder Anwendungen im zweiten Konto (B) verwenden die von S3 Access Grants bereitgestellten Anmeldeinformationen, um auf die S3-Daten in Ihrem Konto (A) zuzugreifen.

- Schritt 1: Konfigurieren Sie eine S3 Access Grants-Instanz in Ihrem Konto, z. B. die Konto-ID111122223333, in der die S3-Daten gespeichert sind.
- Schritt 2: Konfigurieren Sie die Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto, 111122223333 um Zugriff auf das zweite Konto zu gewähren, z. B. die Konto-ID444455556666.
- Schritt 3: Konfigurieren Sie die IAM Berechtigungen für den IAM Principal im zweiten Konto444455556666, um Anmeldeinformationen von der S3 Access Grants-Instanz in Ihrem Konto anzufordern111122223333.
- Schritt 4: Erstellen Sie in Ihrem Konto einen Grant111122223333, der dem IAM Principal im zweiten Konto 444455556666 Zugriff auf einige der S3-Daten in Ihrem Konto gewährt111122223333.

Schritt 1: Konfigurieren Sie eine S3 Access Grants-Instanz in Ihrem Konto

Zunächst benötigen Sie eine S3 Access Grants-Instanz in Ihrem Konto, 111122223333 um den Zugriff auf Ihre Amazon S3 S3-Daten zu verwalten. Sie müssen in jeder Instanz eine S3 Access Grants-Instanz erstellen AWS-Region wo die S3-Daten gespeichert sind, die Sie teilen möchten. Wenn Sie Daten in mehr als einem teilen AWS-Region, wiederholen Sie dann jeden dieser Konfigurationsschritte für jeden AWS-Region. Wenn Sie bereits eine S3 Access Grants-Instanz in der AWS-Region wo Ihre S3-Daten gespeichert sind, fahren Sie mit dem nächsten Schritt fort. Wenn Sie keine S3 Access Grants-Instanz konfiguriert haben, finden Sie hier Informationen [Arbeiten mit S3 Access Grants-Instanzen](#) zum Abschließen dieses Schritts.

Schritt 2: Konfigurieren Sie die Ressourcenrichtlinie für Ihre S3 Access Grants-Instanz, um kontoübergreifenden Zugriff zu gewähren

Nachdem Sie in Ihrem Konto eine S3 Access Grants-Instanz 111122223333 für den kontoübergreifenden Zugriff erstellt haben, konfigurieren Sie die ressourcenbasierte Richtlinie für die S3 Access Grants-Instanz in Ihrem Konto, um kontoübergreifenden Zugriff 111122223333 zu gewähren. Die S3-Access-Grants-Instanz selbst unterstützt ressourcenbasierte Richtlinien. Wenn die richtige ressourcenbasierte Richtlinie vorhanden ist, können Sie Zugriff gewähren für AWS Identity and Access Management (IAM) Benutzer oder Rollen von anderen AWS-Konten zu Ihrer S3 Access Grants-Instanz. Der kontoübergreifende Zugriff gewährt nur diese Berechtigungen (Aktionen):

- `s3:GetAccessGrantsInstanceForPrefix`— Der Benutzer, die Rolle oder die App kann die S3 Access Grants-Instanz abrufen, die ein bestimmtes Präfix enthält.
- `s3:ListAccessGrants`

- `s3:ListAccessLocations`
- `s3:ListCallerAccessGrants`
- `s3:GetDataAccess`— Der Benutzer, die Rolle oder die App kann temporäre Anmeldeinformationen anfordern, die auf dem Zugriff basieren, den Sie über S3 Access Grants erhalten haben. Mit diesen Anmeldeinformationen können Sie auf die S3-Daten zugreifen, für die Ihnen Zugriff gewährt wurde.

Sie können auswählen, welche dieser Berechtigungen in die Ressourcenrichtlinie aufgenommen werden sollen. Diese Ressourcenrichtlinie auf der S3 Access Grants-Instanz ist eine normale ressourcenbasierte Richtlinie und unterstützt alles, was die [IAMRichtliniensprache](#) unterstützt. In derselben Richtlinie können Sie beispielsweise Zugriff auf bestimmte IAM Identitäten in Ihrem Konto gewähren, indem Sie die `aws:PrincipalArn` Bedingung verwenden. Bei S3 Access Grants müssen Sie das jedoch nicht tun. Stattdessen können Sie innerhalb Ihrer S3 Access Grants-Instanz Berechtigungen für einzelne IAM Identitäten von Ihrem Konto aus sowie für das andere Konto erstellen. Indem Sie die einzelnen Zugriffsberechtigungen über S3 Access Grants verwalten, können Sie Ihre Berechtigungen skalieren.

Wenn Sie bereits verwenden [AWS Resource Access Manager](#) (AWS RAM), können Sie es verwenden, um Ihre `s3:AccessGrants` Ressourcen mit anderen Konten oder innerhalb Ihrer Organisation zu teilen. Weitere Informationen findest du unter [Arbeiten mit geteilten AWS](#). Weitere Informationen finden Sie in den Ressourcen. Wenn Sie nicht verwenden AWS RAM, können Sie die Ressourcenrichtlinie auch mithilfe der S3 Access API Grants-Operationen oder der AWS Command Line Interface (AWS CLI).

Verwenden der S3-Konsole

Wir empfehlen Ihnen die Verwendung des AWS Resource Access Manager (AWS RAM) Konsole, um Ihre `s3:AccessGrants` Ressourcen mit anderen Konten oder innerhalb Ihrer Organisation zu teilen. Gehen Sie wie folgt vor, um S3 Access Grants kontoübergreifend zu teilen:

So konfigurieren Sie die Ressourcenrichtlinie für die S3 Access Grants-Instanz:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie AWS-Region aus dem AWS-Region Selektor.
3. Wählen Sie im linken Navigationsbereich Access Grants aus.

4. Wählen Sie auf der Seite Access Grants-Instanz im Abschnitt Instanz in diesem Konto die Option Instanz teilen aus. Dadurch werden Sie weitergeleitet zur AWS RAM Konsole.
5. Wählen Sie „Ressourcenfreigabe erstellen“ aus.
6. Folgen Sie dem AWS RAM Schritte zum Erstellen der Ressourcenfreigabe. Weitere Informationen finden Sie unter Eine Ressourcenfreigabe [erstellen in AWS RAM](#).

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Sie können die Ressourcenrichtlinie mithilfe des `put-access-grants-instance-resource-policy` CLI Befehls hinzufügen.

Wenn Sie dem zweiten Konto kontoübergreifenden Zugriff für die S3 Access Grants-Instanz in Ihrem Konto 111122223333 gewähren möchten444455556666, 111122223333 sollte die Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto dem zweiten Konto die 444455556666 Erlaubnis geben, die folgenden Aktionen auszuführen:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Geben Sie in der Ressourcenrichtlinie für die S3 Access Grants-Instanz den ARN Namen Ihrer S3 Access Grants-Instanz als und das zweite Konto 444455556666 als an. Resource Principal Um das folgende Beispiel zu verwenden, ersetzen Sie das *user input placeholders* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "444455556666"
      },
      "Action": [
```



```

"s3:ListAccessGrants",
"s3:ListAccessGrantsLocations",
"s3:GetDataAccess",
"s3:GetAccessGrantsInstanceForPrefix"
],
"Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}

```

Verwenden Sie den folgenden Befehl, um die Ressourcenrichtlinie für die S3 Access Grants-Instanz hinzuzufügen oder zu aktualisieren. Wenn Sie den folgenden Beispielbefehl verwenden, ersetzen Sie den *user input placeholders* durch Ihre eigenen Informationen.

Example Fügen Sie die Ressourcenrichtlinie für die S3 Access Grants-Instanz hinzu oder aktualisieren Sie sie

```

aws s3control put-access-grants-instance-resource-policy \
--account-id 111122223333 \
--policy file://resourcePolicy.json \
--region us-east-2
{
  "Policy": "{\n
    \"Version\": \"2012-10-17\", \n
    \"Statement\": [{\n
      \"Effect\": \"Allow\", \n
      \"Principal\": {\n
        \"AWS\": \"444455556666\" \n
      }, \n
      \"Action\": [\n
        \"s3:ListAccessGrants\", \n
        \"s3:ListAccessGrantsLocations\", \n
        \"s3:GetDataAccess\", \n
        \"s3:GetAccessGrantsInstanceForPrefix\", \n
        \"s3:ListCallerAccessGrants\" \n
      ], \n
      \"Resource\": \"arn:aws:s3:us-east-2:111122223333:access-grants/default\" \n
    } \n
  ] \n
  } \n",
  "CreatedAt": "2023-06-16T00:07:47.473000+00:00"
}

```

Example Abrufen einer S3-Access-Grants-Ressourcenrichtlinie

Sie können die auch verwendenCLI, um eine Ressourcenrichtlinie für eine S3 Access Grants-Instanz abzurufen oder zu löschen.

Verwenden Sie den folgenden Beispielbefehl, um eine S3 Access Grants-Ressourcenrichtlinie abzurufen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-access-grants-instance-resource-policy \  
--account-id 111122223333 \  
--region us-east-2  
  
{  
  "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\  
\", \"Principal\":{\"AWS\":\"arn:aws:iam:111122223333:root\"}, \"Action\":  
[\"s3:ListAccessGrants\", \"s3:ListAccessGrantsLocations\", \"s3:GetDataAccess\",  
\"s3:GetAccessGrantsInstanceForPrefix\", \"s3:ListCallerAccessGrants\"], \"Resource\":  
\"arn:aws:  
s3:us-east-2:111122223333:access-grants/default\"}]}}\",  
  "CreatedAt": "2023-06-16T00:07:47.473000+00:00"  
}
```

Example Löschen einer S3-Access-Grants-Ressourcenrichtlinie

Verwenden Sie den folgenden Beispielbefehl, um eine S3 Access Grants-Ressourcenrichtlinie zu löschen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-access-grants-instance-resource-policy \  
--account-id 111122223333 \  
--region us-east-2  
  
// No response body
```

Verwenden Sie den REST API

Sie können die Ressourcenrichtlinie hinzufügen, indem Sie den verwenden [PutAccessGrantsInstanceResourcePolicy API](#).

Wenn Sie dem zweiten Konto kontenübergreifenden Zugriff für die S3 Access Grants-Instanz in Ihrem Konto 111122223333 gewähren möchten444455556666, 111122223333 sollte die

Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto dem zweiten Konto die 444455556666 Erlaubnis geben, die folgenden Aktionen auszuführen:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Geben Sie in der Ressourcenrichtlinie für die S3 Access Grants-Instanz den ARN Namen Ihrer S3 Access Grants-Instanz als und das zweite Konto 444455556666 als an. Resource Principal Um das folgende Beispiel zu verwenden, ersetzen Sie das *user input placeholders* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "444455556666"
      },
      "Action": [
        "s3:ListAccessGrants",
        "s3:ListAccessGrantsLocations",
        "s3:GetDataAccess",
        "s3:GetAccessGrantsInstanceForPrefix"
      ],
      "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
    }
  ]
}
```

Sie können dann den verwenden [PutAccessGrantsInstanceResourcePolicy API](#), um die Richtlinie zu konfigurieren.

Informationen zur REST API Unterstützung beim Aktualisieren, Abrufen oder Löschen einer Ressourcenrichtlinie für eine S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [PutAccessGrantsInstanceResourcePolicy](#)

- [GetAccessGrantsInstanceResourcePolicy](#)
- [DeleteAccessGrantsInstanceResourcePolicy](#)

Verwendung der AWS SDKs

In diesem Abschnitt finden Sie AWS SDK-Beispiele dafür, wie Sie Ihre S3 Access Grants-Ressourcenrichtlinie so konfigurieren, dass eine Sekunde gewährt wird AWS Kontozugriff auf einige Ihrer S3-Daten.

Java

Sie können eine Ressourcenrichtlinie hinzufügen, aktualisieren, abrufen oder löschen, um den kontoübergreifenden Zugriff für Ihre S3-Access-Grants-Instance zu verwalten.

Example Fügen Sie eine Ressourcenrichtlinie für S3 Access Grants Instance hinzu oder aktualisieren Sie sie

Wenn Sie kontoübergreifenden Zugriff für die S3 Access Grants-Instanz in Ihrem Konto auf das zweite Konto 111122223333 gewähren möchten 444455556666, 111122223333 sollte die Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto dem zweiten Konto die 444455556666 Erlaubnis geben, die folgenden Aktionen auszuführen:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Geben Sie in der Ressourcenrichtlinie für die S3 Access Grants-Instanz den ARN Namen Ihrer S3 Access Grants-Instanz als und das zweite Konto 444455556666 als an. Resource Principal
Um das folgende Beispiel zu verwenden, ersetzen Sie das *user input placeholders* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "444455556666"
      }
    }
  ]
}
```

```

},
"Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
],
"Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}

```

Verwenden Sie das folgende Codebeispiel, um eine Ressourcenrichtlinie für eine S3 Access Grants-Instanz hinzuzufügen oder zu aktualisieren:

```

public void putAccessGrantsInstanceResourcePolicy() {
    PutAccessGrantsInstanceResourcePolicyRequest putRequest =
    PutAccessGrantsInstanceResourcePolicyRequest.builder()
    .accountId(111122223333)
    .policy(RESOURCE_POLICY)
    .build();
    PutAccessGrantsInstanceResourcePolicyResponse putResponse =
    s3Control.putAccessGrantsInstanceResourcePolicy(putRequest);
    LOGGER.info("PutAccessGrantsInstanceResourcePolicyResponse: " + putResponse);
}

```

Antwort:

```

PutAccessGrantsInstanceResourcePolicyResponse(
  Policy={
    "Version": "2012-10-17",
    "Statement": [{
      "Effect": "Allow",
      "Principal": {
        "AWS": "444455556666"
      },
      "Action": [
        "s3:ListAccessGrants",
        "s3:ListAccessGrantsLocations",
        "s3:GetDataAccess",
        "s3:GetAccessGrantsInstanceForPrefix",
        "s3:ListCallerAccessGrants"
      ],
      "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
    }
  ]
}

```

```

    ]]
  }
)

```

Example Abrufen einer S3-Access-Grants-Ressourcenrichtlinie

Verwenden Sie das folgende Codebeispiel, um eine S3 Access Grants-Ressourcenrichtlinie abzurufen. Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```

public void getAccessGrantsInstanceResourcePolicy() {
    GetAccessGrantsInstanceResourcePolicyRequest getRequest =
    GetAccessGrantsInstanceResourcePolicyRequest.builder()
    .accountId(111122223333)
    .build();
    GetAccessGrantsInstanceResourcePolicyResponse getResponse =
    s3Control.getAccessGrantsInstanceResourcePolicy(getRequest);
    LOGGER.info("GetAccessGrantsInstanceResourcePolicyResponse: " + getResponse);
}

```

Antwort:

```

GetAccessGrantsInstanceResourcePolicyResponse(
    Policy={"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
    {"AWS":"arn:aws:iam:444455556666:root"},"Action":
    ["s3:ListAccessGrants","s3:ListAccessGrantsLocations","s3:GetDataAccess","s3:GetAccessGrants
east-2:111122223333:access-grants/default"]]}],
    CreatedAt=2023-06-15T22:54:44.319Z
)

```

Example Löschen einer S3-Access-Grants-Ressourcenrichtlinie

Verwenden Sie das folgende Codebeispiel, um eine S3 Access Grants-Ressourcenrichtlinie zu löschen. Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```

public void deleteAccessGrantsInstanceResourcePolicy() {
    DeleteAccessGrantsInstanceResourcePolicyRequest deleteRequest =
    DeleteAccessGrantsInstanceResourcePolicyRequest.builder()
    .accountId(111122223333)
    .build();
}

```

```
DeleteAccessGrantsInstanceResourcePolicyResponse deleteResponse =
s3Control.putAccessGrantsInstanceResourcePolicy(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResourcePolicyResponse: " + deleteResponse);
}
```

Antwort:

```
DeleteAccessGrantsInstanceResourcePolicyResponse()
```

Schritt 3: Erteilen Sie IAM Identitäten in einem zweiten Konto die Erlaubnis, die S3 Access Grants-Instanz in Ihrem Konto aufzurufen

Nachdem der Eigentümer der Amazon S3 S3-Daten die kontoübergreifende Richtlinie für die S3 Access Grants-Instance im Konto konfiguriert hat111122223333, 444455556666 muss der Besitzer des zweiten Kontos eine identitätsbasierte Richtlinie für seine IAM Benutzer oder Rollen erstellen, und der Eigentümer muss ihnen Zugriff auf die S3 Access Grants-Instance gewähren. Nehmen Sie in die identitätsbasierte Richtlinie eine oder mehrere der folgenden Aktionen auf, je nachdem, was in der Ressourcenrichtlinie für die S3 Access Grants Instance gewährt wurde und welche Berechtigungen Sie gewähren möchten:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`
- `s3:ListCallerAccessGrants`

Im Anschluss an die [AWS Bei kontoübergreifendem Zugriffsmuster](#) 444455556666 müssen die IAM Benutzer oder Rollen im zweiten Konto explizit über eine oder mehrere dieser Berechtigungen verfügen. Erteilen Sie beispielsweise die `s3:GetDataAccess` Berechtigung, damit der IAM Benutzer oder die Rolle die S3 Access Grants-Instanz im Konto aufrufen kann111122223333, um Anmeldeinformationen anzufordern.

Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
{
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "s3:GetDataAccess",  
    ],  
    "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"  
  }  
]
```

Informationen zur Bearbeitung IAM identitätsbasierter Richtlinien finden Sie unter [Bearbeiten von IAM Richtlinien](#) im AWS Identity and Access Management Leitfaden.

Schritt 4: Erstellen Sie in der S3 Access Grants-Instanz Ihres Kontos einen Grant, der der IAM Identität im zweiten Konto Zugriff auf einige Ihrer S3-Daten gewährt

Für den letzten Konfigurationsschritt können Sie in der S3 Access Grants-Instanz in Ihrem Konto 111122223333 einen Grant erstellen, der Zugriff auf die IAM Identität im zweiten Konto 444455556666 auf einige der S3-Daten in Ihrem Konto gewährt. Sie können dies mithilfe der Amazon S3 S3-Konsole, CLI-API, und tunSDKs. Weitere Informationen finden Sie unter [Erstellen von Gewährungen](#).

Geben Sie im Zuschuss Folgendes an AWS ARN der IAM Identität des zweiten Kontos und geben Sie an, für welchen Speicherort in Ihren S3-Daten (ein Bucket, ein Präfix oder ein Objekt) Sie Zugriff gewähren. Dieser Standort muss bereits bei Ihrer S3 Access Grants-Instanz registriert sein. Weitere Informationen finden Sie unter [Registrieren eines Speicherorts](#). Sie können optional ein Unterpräfix angeben. Wenn es sich bei dem Speicherort, für den Sie Zugriff gewähren, um einen Bucket handelt und Sie den Zugriff weiter auf ein bestimmtes Objekt in diesem Bucket einschränken möchten, übergeben Sie den Objektschlüsselnamen in das `S3SubPrefix` Feld. Oder wenn Sie den Zugriff auf die Objekte im Bucket mit Schlüsselnamen einschränken möchten, die mit einem bestimmten Präfix beginnen, wie z. B. `2024-03-research-results/`, dann übergeben `S3SubPrefix=2024-03-research-results/`.

Im Folgenden finden Sie einen CLI Beispielbefehl für die Erstellung einer Zugriffsberechtigung für eine Identität im zweiten Konto. Weitere Informationen finden Sie unter [Erstellen von Gewährungen](#). Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-access-grant \
```



```
--account-id 111122223333 \  
--access-grants-location-id default \  
--access-grants-location-configuration S3SubPrefix=prefixA* \  
--permission READ \  
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::444455556666:role/data-  
consumer-1
```

Nach der Konfiguration des kontoübergreifenden Zugriffs kann der Benutzer oder die Rolle im zweiten Konto Folgendes tun:

- Ruft `ListAccessGrantsInstances` auf, um die S3 Access Grants-Instanzen aufzulisten, die mit dem Programm geteilt wurden AWS RAM. Weitere Informationen finden Sie unter [Rufen Sie die Details einer S3 Access Grants-Instance ab](#).
- Fordert temporäre Anmeldeinformationen von S3 Access Grants an. Weitere Informationen darüber, wie Sie diese Anfragen stellen können, finden Sie unter [Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants](#).

Die Verwendung von AWS Tags mit S3 Access Grants

Tags in Amazon S3 Access Grants besitzen ähnliche Eigenschaften wie [Objekt-Tags](#) in Amazon S3. Jeder Tag ist ein Schlüssel/Wert-Paar. Die Ressourcen in S3 Access Grants, die Sie markieren können, sind [Instances](#), [Speicherorte](#) und [Gewährungen](#).

Note

Das Tagging in S3 Access Grants verwendet andere API Operationen als das Tagging von Objekten. S3 Access Grants verwendet die [TagResource](#), [UntagResource](#), und [ListTagsForResource](#) API Operationen, bei denen es sich bei einer Ressource entweder um eine S3 Access Grants-Instanz, einen registrierten Standort oder um eine Zugriffsberechtigung handeln kann.

Ähnlich wie bei [Objekt-Tags](#) gelten die folgenden Einschränkungen:

- Sie können Tags zu neuen S3-Access-Grants-Ressourcen hinzufügen, wenn Sie diese erstellen, oder zu vorhandenen Ressourcen hinzufügen.
- Sie können einer Ressource bis zu 10 Tags zuordnen. Wenn mehrere Tags derselben Ressource zugeordnet sind, müssen sie eindeutige Tag-Schlüssel besitzen.

- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Tags werden intern in UTF -16 dargestellt. In UTF -16 verbrauchen Zeichen entweder 1 oder 2 Zeichenpositionen.
- Schlüssel und Werte unterscheiden nach Groß- und Kleinschreibung.

Weitere Informationen zu Tag-Einschränkungen finden Sie unter [Einschränkungen für benutzerdefinierte Tags](#) im AWS Billing Benutzerleitfaden.

Sie können Ressourcen in S3 Access Grants taggen, indem Sie den AWS Command Line Interface (AWS CLI), der Amazon S3 REST API oder der AWS SDKs.

Verwendung der AWS CLI

Um das zu installieren AWS CLI, siehe [Installation des AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.

Sie können eine S3-Access-Grants-Ressource während oder nach der Erstellung markieren. Die folgenden Beispiele zeigen, wie Sie eine S3-Access-Grants-Instance markieren oder eine vorhandene Markierung aufheben. Sie können ähnliche Operationen für registrierte Speicherorte und Zugriffsgewährungen durchführen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example – Erstellen einer S3-Access-Grants-Instance mit Tags

```
aws s3control create-access-grants-instance \  
  --account-id 111122223333 \  
  --profile access-grants-profile \  
  --region us-east-2 \  
  --tags Key=tagKey1,Value=tagValue1
```

Antwort:

```
{  
  "CreatedAt": "2023-10-25T01:09:46.719000+00:00",  
  "AccessGrantsInstanceId": "default",  
  "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/  
default"
```

```
}
```

Example – Markieren einer bereits erstellten S3-Access-Grants-Instance

```
aws s3control tag-resource \  
--account-id 111122223333 \  
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \  
--profile access-grants-profile \  
--region us-east-2 \  
--tags Key=tagKey2,Value=tagValue2
```

Example – Auflisten der Tags für eine S3-Access-Grants-Instance

```
aws s3control list-tags-for-resource \  
--account-id 111122223333 \  
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \  
--profile access-grants-profile \  
--region us-east-2
```

Antwort:

```
{  
  "Tags": [  
    {  
      "Key": "tagKey1",  
      "Value": "tagValue1"  
    },  
    {  
      "Key": "tagKey2",  
      "Value": "tagValue2"  
    }  
  ]  
}
```

Example – Aufheben der Markierung für eine S3-Access-Grants-Instance

```
aws s3control untag-resource \  
--account-id 111122223333 \  
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \  
--profile access-grants-profile \  
--region us-east-2 \  

```

```
--tag-keys "tagKey2"
```

Mit dem REST API

Sie können Amazon S3 verwenden, API um Tags für eine S3 Access Grants-Instance, einen registrierten Standort oder eine Zugriffserteilung zu taggen, aufzuheben oder aufzulisten.

Informationen zur REST API Unterstützung der Verwaltung von S3 Access Grants-Tags finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [TagResource](#)
- [UntagResource](#)
- [ListTagsForResource](#)

Limits für S3 Access Grants

Für [S3 Access Grants](#) gelten die folgenden Limits:

Note

Wenn Ihr Anwendungsfall diese Einschränkungen überschreitet, [wenden Sie sich an AWS Support](#), um höhere Grenzwerte anzufordern.

S3-Access-Grants-Instance

Sie können 1 S3 Access Grants-Instanz pro erstellen AWS-Region pro Konto. Siehe [Erstellen einer S3-Access-Grants-Instance](#).

S3-Access-Grants Speicherort

Sie können pro S3-Access-Grants-Instance 1.000 S3-Access-Grants-Speicherorte registrieren. Siehe [Registrieren eines S3-Access-Grants-Speicherorts](#).

Gewährung

Sie können pro S3-Access-Grants-Instance 100.000 Gewährungen erstellen. Siehe [Erstellen einer Gewährung](#).

S3-Access-Grants-Integrationen

S3 Access Grants können wie folgt verwendet werden AWS Dienste und Funktionen. Diese Seite wird aktualisiert, wenn neue Integrationen verfügbar sind.

Amazon Athena

[Verwendung von IAM Identity Center-fähigen Athena-Arbeitsgruppen](#)

Amazon EMR

[Starten Sie einen EMR Amazon-Cluster mit S3 Access Grants](#)

Amazon EMR auf EKS

[Starten Sie einen EMR EKS Amazon-On-Cluster mit S3 Access Grants](#)

EMR Serverless-Anwendung von Amazon

[Starten Sie eine Amazon EMR Serverless-Anwendung mit S3 Access Grants](#)

AWS IAM Identity Center

[Vertrauenswürdige Identitätsverteilung zwischen Anwendungen](#)

Amazon SageMaker Studio

[Verwenden von Amazon S3 Access Grants mit Amazon SageMaker Studio und dem SDK for Python \(Boto3\) -Plugin](#)

Die Verwendung von S3 Access Grants in Amazon SageMaker Studio-Notebooks ist jetzt einfacher, wenn Sie das SDK for Python (Boto3) -Plugin verwenden. Richten Sie Zugriffsberechtigungen für IAM Principals ein und AWS IAM Identity Center Verzeichnisbenutzer, vorher. Amazon SageMaker Studio unterstützt zwar nicht nativ Identity Provider-Verzeichnisbenutzer, aber Sie können benutzerdefinierten Python-Code schreiben, indem Sie das Plug-in verwenden, das diesen Identitäten den Zugriff auf S3-Daten über S3 Access Grants ermöglicht. Der Datenzugriff erfolgt mit Hilfe des Plugins und nicht über Amazon SageMaker.

Open-Source-Python-Frameworks

[Amazon S3 Access Grants lässt sich jetzt in Open-Source-Python-Frameworks integrieren](#)

Verwaltung des Zugriffs mit ACLs

Zugriffskontrolllisten (ACLs) sind eine der ressourcenbasierten Optionen, mit denen Sie den Zugriff auf Ihre Buckets und Objekte verwalten können. Sie können sie verwenden ACLs, um anderen Benutzern grundlegende Lese-/Schreibberechtigungen zu gewähren AWS-Konten. Es gibt Einschränkungen bei der Verwaltung von Berechtigungen mithilfe von ACLs.

Beispielsweise können Sie Berechtigungen nur anderen Personen gewähren AWS-Konten; Sie können Benutzern in Ihrem Konto keine Berechtigungen gewähren. Sie können keine bedingten Berechtigung erteilen, und Sie können nicht explizit Berechtigungen verweigern. ACLs sind für bestimmte Szenarien geeignet. Zum Beispiel, wenn ein Bucket-Besitzer andere erlaubt AWS-Konten um Objekte hochzuladen, können die Berechtigungen für diese Objekte nur mithilfe ACL von object by verwaltet werden AWS-Konto dem das Objekt gehört.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner enforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Weitere Informationen zu ACLs finden Sie in den folgenden Themen.

Themen

- [Übersicht über die Zugriffskontrollliste \(ACL\)](#)
- [Konfiguration ACLs](#)
- [Beispiele für Richtlinien für ACLs](#)

Übersicht über die Zugriffskontrollliste (ACL)

Mit Amazon S3 S3-Zugriffskontrolllisten (ACLs) können Sie den Zugriff auf Buckets und Objekte verwalten. Jedem Bucket und Objekt ist eine Unterressource ACL angehängt. Es definiert welche AWS-Konten oder Gruppen, denen Zugriff gewährt wird, und die Art des Zugriffs. Wenn eine Anfrage für eine Ressource eingeht, überprüft Amazon S3 die entsprechende Anfrage, ACL um sicherzustellen, dass der Anforderer über die erforderlichen Zugriffsberechtigungen verfügt.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner enforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Wenn Sie einen Bucket oder ein Objekt erstellen, erstellt Amazon S3 eine Standardeinstellung ACL, die dem Eigentümer der Ressource die volle Kontrolle über die Ressource gewährt. Dies wird im folgenden Beispiel-Bucket gezeigt ACL (das Standardobjekt ACL hat dieselbe Struktur):

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="Canonical User">
        <ID>*** Owner-Canonical-User-ID ***</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Das Beispiel ACL enthält ein `Owner` Element, das den Besitzer anhand der AWS-Konto ist die kanonische Benutzer-ID. Anweisungen zum Auffinden Ihrer kanonischen Benutzer-ID finden Sie unter [Suche nach einem AWS-Konto kanonische Benutzer-ID](#). Das `Grant` Element identifiziert den Empfänger (entweder ein AWS-Konto oder eine vordefinierte Gruppe) und die erteilte Genehmigung. Diese Standardeinstellung ACL hat ein `Grant` Element für den Besitzer. Sie erteilen Berechtigungen, indem Sie `Grant`-Elemente hinzufügen, wobei jedes Recht den Empfänger und die Berechtigung identifiziert.

Note

Ein ACL kann bis zu 100 Zuschüsse haben.

Themen

- [Wer ist ein Empfänger?](#)
- [Welche Berechtigungen kann ich erteilen?](#)

- [aclRequired-Werte für allgemeine Amazon-S3-Anfragen](#)
- [Stichprobe ACL](#)
- [In Dosen ACL](#)

Wer ist ein Empfänger?

Ein Stipendiat kann ein sein AWS-Konto oder eine der vordefinierten Amazon S3 S3-Gruppen. Sie erteilen die Erlaubnis einem AWS-Konto mit der E-Mail-Adresse oder der kanonischen Benutzer-ID. Wenn Sie jedoch in Ihrem Zuschussantrag eine E-Mail-Adresse angeben, findet Amazon S3 die kanonische Benutzer-ID für dieses Konto und fügt sie dem hinzu. ACL Die Ergebnisse enthalten ACLs immer die kanonische Benutzer-ID für AWS-Konto, nicht die E-Mail-Adresse des AWS-Konto.

Wenn Sie Zugriffsrechte erteilen, geben Sie jeden Empfänger als *type*="value"-Paar an, wobei *type* einer der folgenden ist:

- *id*— Wenn der angegebene Wert die kanonische Benutzer-ID eines ist AWS-Konto
- *uri* – Wenn Sie einer vordefinierten Gruppe Berechtigungen erteilen
- *emailAddress*— Wenn der angegebene Wert die E-Mail-Adresse eines ist AWS-Konto

Important

Die Verwendung von E-Mail-Adressen zur Angabe eines Empfängers wird nur in den folgenden Fällen unterstützt AWS Regionen:

- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Europa (Irland)
- Südamerika (São Paulo)

Eine Liste aller von Amazon S3 unterstützten Regionen und Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz.

Example Beispiel: E-Mail-Adresse

Der folgende `x-amz-grant-read` Header gewährt beispielsweise AWS-Konten anhand von E-Mail-Adressen identifizierte Berechtigungen zum Lesen von Objektdaten und deren Metadaten:

```
x-amz-grant-read: emailAddress="xyz@example.com", emailAddress="abc@example.com"
```

Warning

Wenn Sie andere gewähren AWS-Konten Zugriff auf Ihre Ressourcen, beachten Sie, dass AWS-Konten können ihre Berechtigungen an Benutzer unter ihren Konten delegieren. Man spricht auch von einem kontoübergreifenden Zugriff. Informationen zur Verwendung des kontoübergreifenden Zugriffs finden Sie unter [Erstellen einer Rolle zur Delegierung von Berechtigungen an einen IAM Benutzer im IAM Benutzerhandbuch](#).

Suche nach einem AWS-Konto kanonische Benutzer-ID

Die kanonische Benutzer-ID ist mit Ihrer verknüpft AWS-Konto. Diese ID besteht aus einer langen Zeichenfolge, z. B.:

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Informationen darüber, wie Sie die kanonische Benutzer-ID für Ihr Konto finden, finden Sie unter [Finden Sie die kanonische Benutzer-ID für Ihr AWS-Konto](#) in der AWS Referenzhandbuch zur Kontoverwaltung.

Sie können auch die kanonische Benutzer-ID eines nachschlagen AWS-Konto indem Sie die ACL eines Buckets oder eines Objekts lesen, zu dem AWS-Konto hat Zugriffsberechtigungen. Wenn eine Einzelperson AWS-Konto denen durch eine Erteilungsanfrage Berechtigungen erteilt wurden, wird dem ein Eintrag ACL mit der kanonischen Benutzer-ID des Kontos hinzugefügt.

 Note


Falls Sie Ihren Bucket öffentlich machen (nicht empfohlen), können beliebige, nicht authentifizierte Benutzer Objekte in den Bucket hochladen. Diese anonymen Benutzer haben kein AWS-Konto. Wenn ein anonymes Benutzer ein Objekt in Ihren Bucket hochlädt, fügt Amazon S3 eine spezielle kanonische Benutzer-ID (65a011a29cdf8ec533ec3d1ccaae921c) als Objekteigentümer in der hinzu. ACL Weitere Informationen finden Sie unter [Amazon-S3-Bucket- und Objekt-Eigentümerschaft](#).

Vordefinierte Gruppen in Amazon S3

Amazon S3 besitzt mehrere vordefinierte Gruppen. Wenn Sie einer Gruppe Kontozugriff gewähren, geben Sie eine der Amazon S3 URIs anstelle einer kanonischen Benutzer-ID an. Amazon S3 stellt die folgenden vordefinierten Gruppen bereit:

- Gruppe „Authentifizierte Benutzer“ – Repräsentiert durch `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`.


Diese Gruppe steht für alle AWS-Konten. Die Zugriffsberechtigung für diese Gruppe erlaubt jede AWS-Konto um auf die Ressource zuzugreifen. Alle Anfragen müssen jedoch signiert (authentifiziert) sein.

 Warning

Wenn Sie der Gruppe „Authentifizierte Benutzer“ Zugriff gewähren, können alle AWS Ein authentifizierter Benutzer auf der ganzen Welt kann auf Ihre Ressource zugreifen.

- Gruppe „Alle Benutzer“ – Repräsentiert durch `http://acs.amazonaws.com/groups/global/AllUsers`.

Die Zugriffsberechtigung für diese Gruppe gestattet jedem, auf die Ressource zuzugreifen. Die Anfragen können signiert (authentifiziert) oder nicht signiert (anonym) sein. Nicht signierte Anfragen lassen den Authentifizierungs-Header in der Anfrage weg.

 Warning

Wir empfehlen dringend, dass Sie nie der Gruppe Alle Benutzer WRITE-, WRITE_ACP- oder FULL_CONTROL-Berechtigungen erteilen. Während WRITE-Berechtigungen es

Nichtbesitzern beispielsweise nicht erlauben, vorhandene Objekte zu überschreiben oder zu löschen, erlauben WRITE-Berechtigungen jedem, Objekte in Ihrem Bucket zu speichern, für die Ihnen eine Rechnung gestellt wird. Weitere Informationen zu diesen Berechtigungen finden Sie im folgenden Abschnitt [Welche Berechtigungen kann ich erteilen?](#).

- Gruppe „Protokollbereitstellung“ – Repräsentiert durch `http://acs.amazonaws.com/groups/s3/LogDelivery`.

Die WRITE-Berechtigung für einen Bucket gestattet dieser Gruppe, Serverzugriff-Protokolle (siehe [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#)) in den Bucket zu schreiben.

Note

Bei der Nutzung ACLs kann ein Stipendiat ein AWS-Konto oder eine der vordefinierten Amazon S3 S3-Gruppen. Der Empfänger kann jedoch kein IAM Benutzer sein. Weitere Informationen zur AWS Benutzer und darin enthaltene Berechtigungen IAM, siehe Verwenden [AWS Identity and Access Management](#).

Welche Berechtigungen kann ich erteilen?

In der folgenden Tabelle sind die Berechtigungen aufgeführt, die Amazon S3 in einem unterstützt ACL. Der Satz von ACL Berechtigungen ist für ein Objekt ACL und einen Bucket identisch ACL. Je nach Kontext (Bucket ACL oder Objekt ACL) gewähren diese ACL Berechtigungen jedoch Berechtigungen für bestimmte Buckets oder Objektoperationen. Die Tabelle listet die Berechtigungen auf und beschreibt, was sie im Kontext der Objekte und Buckets bedeuten.

Weitere Informationen zu ACL Berechtigungen in der Amazon S3 S3-Konsole finden Sie unter [Konfiguration ACLs](#).

ACL Berechtigungen

Berechtigung	Bei Rechteerteilung für einen Bucket	Bei Rechteerteilung für ein Objekt
READ	Gestattet dem Empfänger, die Objekte im Bucket aufzulisten	Gestattet dem Empfänger, die Objektdaten und seine Metadaten zu lesen

Berechtigung	Bei Rechteerteilung für einen Bucket	Bei Rechteerteilung für ein Objekt
WRITE	Gestattet dem Empfänger, neue Objekte im Bucket zu erstellen. Für die Bucket- und Objekteigentümer vorhandener Objekte können auch Löschungen und Überschreibungen dieser Objekte ermöglicht werden.	Nicht zutreffend
READ_ACP	Ermöglicht dem Empfänger, den Bucket zu lesen ACL	Ermöglicht dem Empfänger, das Objekt zu lesen ACL
WRITE_ACP	Ermöglicht dem Empfänger, das ACL für den entsprechenden Bucket zu schreiben	Ermöglicht dem Empfänger, das ACL für das entsprechende Objekt zu schreiben
FULL_CONTROL	Gewährt dem Berechtigungsempfänger die READ-, WRITE-, READ_ACP- und WRITE_ACP - Berechtigungen für den Bucket	Gewährt dem Berechtigungsempfänger die READ-, READ_ACP- und WRITE_ACP -Berechtigungen für das Objekt

Warning

Seien Sie beim Gewähren von Zugriffsberechtigungen auf Ihre S3-Buckets und -Objekte vorsichtig. Beispielsweise kann der Berechtigungsempfänger nach dem Gewähren des WRITE-Zugriffs auf einen Bucket Objekte im Bucket erstellen. Wir empfehlen dringend, dass Sie vor dem Erteilen von Berechtigungen den gesamten Abschnitt zu [Übersicht über die Zugriffskontrollliste \(ACL\)](#) lesen.

Zuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen

Wie in der obigen Tabelle dargestellt, ACL erlaubt an nur eine begrenzte Anzahl von Berechtigungen, verglichen mit der Anzahl von Berechtigungen, die Sie in einer Zugriffsrichtlinie festlegen können (siehe [Politische Maßnahmen für Amazon S3](#)). Jede dieser Berechtigungen erlaubt einen oder mehrere Amazon-S3-Vorgänge.

Die folgende Tabelle zeigt, wie die einzelnen ACL Berechtigungen den entsprechenden Zugriffsrichtlinienberechtigungen zugeordnet werden. Wie Sie sehen können, erlaubt die Zugriffsrichtlinie mehr Berechtigungen als ACL jede. Sie werden ACLs hauptsächlich verwendet, um grundlegende Lese-/Schreibberechtigungen zu gewähren, ähnlich wie Dateisystemberechtigungen. Weitere Informationen darüber, wann Sie eine verwenden sollten ACL, finden Sie unter [Identity and Access Management für Amazon S3](#)

Weitere Informationen zu ACL Berechtigungen in der Amazon S3 S3-Konsole finden Sie unter [Konfiguration ACLs](#).

ACLErlaubnis	Entsprechende Zugriffsrichtlinien-Berechtigungen, wenn die ACL Erlaubnis für einen Bucket erteilt wird	Entsprechende Zugriffsrichtlinienberechtigungen, wenn die ACL Berechtigung für ein Objekt erteilt wird
READ	<code>s3:ListBucket</code> , <code>s3:ListBucketVersions</code> und <code>s3:ListBucketMultipartUploads</code>	<code>s3:GetObject</code> und <code>s3:GetObjectVersion</code>
WRITE	<p><code>s3:PutObject</code></p> <p>Der Bucket-Eigentümer kann jedes Objekt im Bucket erstellen, überschreiben und löschen, und der Objekteigentümer hat FULL_CONTROL über sein Objekt.</p> <p>Wenn der Empfänger der Bucket-Besitzer ist, ACL ermöglicht die Erteilung von WRITE Berechtigungen in einem Bucket außerdem, dass die <code>s3:DeleteObjectVersion</code> Aktion für jede Version in diesem Bucket ausgeführt werden kann.</p>	Nicht zutreffend
READ_ACP	<code>s3:GetBucketAcl</code>	<code>s3:GetObjectAcl</code> und <code>s3:GetObjectVersionAcl</code>

ACLErlaubnis	Entsprechende Zugriffsrichtlinie n-Berechtigungen, wenn die ACL Erlaubnis für einen Bucket erteilt wird	Entsprechende Zugriffsrichtlinie nberechtigungen, wenn die ACL Berechtigung für ein Objekt erteilt wird
WRITE_ACP	s3:PutBucketAcl	s3:PutObjectAcl und s3:PutObjectVersionAcl
FULL_CONTROL	Entspricht der Erteilung von READWRITE, READ_ACP, und WRITE_ACP ACL Berechtigungen. Dementsprechend entspricht diese ACL Berechtigung einer Kombination aus entsprechenden Zugriffsrichtlinie nberechtigungen.	Entspricht der Gewährung von READ_ACP, und WRITE_ACP ACL Berechtigungen. Dementsprechend entspricht diese ACL Berechtigung einer Kombination aus entsprechenden Zugriffsrichtlinie nberechtigungen.

Bedingungsschlüssel

Wenn Sie Zugriffsrichtlinienberechtigungen gewähren, können Sie Bedingungsschlüssel verwenden, um den Wert für ein Objekt mithilfe einer Bucket-Richtlinie einzuschränken. Die folgenden Kontextschlüssel entsprechen ACLs. Sie können diese Kontextschlüssel verwenden, um die Verwendung eines bestimmten Elements ACL in einer Anfrage vorzuschreiben:

- s3:x-amz-grant-read - Erfordert Lesezugriff.
- s3:x-amz-grant-write - Erfordert Schreibzugriff.
- s3:x-amz-grant-read-acp- Lesezugriff auf den Bucket erforderlichACL.
- s3:x-amz-grant-write-acp- Schreibzugriff auf den Bucket erforderlichACL.
- s3:x-amz-grant-full-control - Erfordert vollständige Kontrolle.
- s3:x-amz-acl - Erfordert eine [In Dosen ACL](#).

Richtlinien, die ACL -spezifische Header beinhalten, finden Sie beispielsweise unter [Gewährung von PutObject s3-Berechtigungen mit einer Bedingung, die voraussetzt, dass der Bucket-Besitzer die volle Kontrolle erhält](#). Eine vollständige Liste der Amazon S3-spezifischen Bedingungsschlüssel finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

ac1Required-Werte für allgemeine Amazon-S3-Anfragen

Um Amazon S3 S3-Anfragen zu identifizieren, die ACLs für eine Autorisierung erforderlich sind, können Sie den `ac1Required` Wert in den Amazon S3 S3-Serverzugriffsprotokollen verwenden oder AWS CloudTrail. Der `ac1Required` Wert, der in CloudTrail unseren Amazon S3-Serverzugriffsprotokollen erscheint, hängt davon ab, welche Operationen aufgerufen wurden, und von bestimmten Informationen über den Anforderer, Objekteigentümer und Bucket-Besitzer. Wenn keine erforderlich ACLs waren, oder wenn Sie die Option „Gespeichertbucket-owner-full-control“ ACL festlegen oder wenn die Anfragen gemäß Ihrer Bucket-Richtlinie zulässig sind, lautet die `ac1Required` Wertzeichenfolge in den Amazon S3-Serverzugriffsprotokollen - "" und fehlt in CloudTrail.

In den folgenden Tabellen sind die erwarteten `ac1Required` Werte in CloudTrail unseren Amazon S3 S3-Serverzugriffsprotokollen für die verschiedenen Amazon S3 API S3-Operationen aufgeführt. Sie können diese Informationen verwenden, um zu verstehen, von welchen Amazon S3 S3-Vorgängen ACLs die Autorisierung abhängt. In den folgenden Tabellen stellen A, B und C die verschiedenen Konten dar, die dem Anforderer, Objekteigentümer und Bucket-Besitzer zugeordnet sind. Einträge mit einem Sternchen (*) geben eines der Konten A, B oder C an.

Note

PutObject Die Operationen in der folgenden Tabelle weisen, sofern nicht anders angegeben, auf Anfragen hin, für die kein Wert festgelegt wurde ACL, sofern es sich nicht um einen ACL handelt bucket-owner-full-control ACL. Ein Nullwert für `ac1Required` gibt an, dass dieser Wert `ac1Required` fehlt in AWS CloudTrail Protokolle.

aclRequired Werte für CloudTrail


Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequired Wert	Grund
GetObject	A	A	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto
	A	B	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto mit erzwungen em Bucket- Eigentümer
	A	A	B	Ja	Null	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	A	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
	A	A	B	Ja	Null	Von Bucket- Richtlinie gewährter

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
						kontoüber greifender Zugriff
	A	B	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
	A	B	C	Ja	Null	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	B	C	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
PutObject	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
	A	Nicht zutreffend	B	Ja	Null	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	Nicht zutreffend	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
PutObject mit einem ACL (außerbucket owner- full- control)	*	Nicht zutreffend	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL
ListObject s	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
	A	Nicht zutreffend	B	Ja	Null	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	Nicht zutreffend	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
DeleteObj ect	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto
	A	Nicht zutreffend	B	Ja	Null	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
	A	Nicht zutreffend	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
PutObject Ac1	*	*	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL
PutBucket Ac1	*	Nicht zutreffend	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL

 Note

REST.PUT.OBJECT Die Operationen in der folgenden Tabelle weisen, sofern nicht anders angegeben, auf Anfragen hin, die keinen Wert festlegen ACL, es sei denn, der ACL ist ein bucket-owner-full-control ACL. Eine aclRequired-Wertzeichenfolge von „-“ gibt einen Nullwert in den Amazon-S3-Serverzugriffsprotokollen an.

aclRequired-Werte für Amazon-S3-Server-Zugriffsprotokolle

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequired Wert	Grund
REST.GET. OBJECT	A	A	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto
	A	B	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto mit erzwungen em Bucket- Eigentümer
	A	A	B	Ja	-	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	A	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
	A	B	B	Ja	-	Von Bucket- Richtlinie gewährter

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
						kontoüber greifender Zugriff
	A	B	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
	A	B	C	Ja	-	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	B	C	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
REST.PUT. OBJECT	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
	A	Nicht zutreffend	B	Ja	-	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	Nicht zutreffend	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
REST.PUT. OBJECT mit einem ACL (außerbucke owner- full- control)	*	Nicht zutreffend	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL
REST.GET. BUCKET	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
	A	Nicht zutreffend	B	Ja	-	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff
	A	Nicht zutreffend	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
REST .DELE TE .OBJECT	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto
	A	Nicht zutreffend	B	Ja	-	Von Bucket- Richtlinie gewährter kontoüber greifender Zugriff

Vorgangsn ame	Auftragge ber	Objekteig entümer	Bucket-Ei gentümer	Bucket- Richtlinie gewährt Zugriff	aclRequir ed Wert	Grund
	A	Nicht zutreffend	B	Nein	Ja	Der kontenübe rgreifende Zugriff ist abhängig von ACL
REST.PUT. ACL	*	*	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL

Stichprobe ACL

Das folgende Beispiel ACL für einen Bucket identifiziert den Eigentümer der Ressource und eine Reihe von Zuschüssen. Das Format ist die XML Darstellung von ACL in Amazon S3 RESTAPI. Der Bucket-Eigentümer hat die FULL_CONTROL über die Ressource. Darüber hinaus ACL zeigt das, wie Berechtigungen für eine Ressource an zwei Personen erteilt werden AWS-Konten, identifiziert durch die kanonische Benutzer-ID und zwei der vordefinierten Amazon S3 S3-Gruppen, die im vorherigen Abschnitt besprochen wurden.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>owner-canonical-user-ID</ID>
    <DisplayName>display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>owner-canonical-user-ID</ID>
```

```

    <DisplayName>display-name</DisplayName>
  </Grantee>
  <Permission>FULL_CONTROL</Permission>
</Grant>

<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
    <ID>user1-canonical-user-ID</ID>
    <DisplayName>display-name</DisplayName>
  </Grantee>
  <Permission>WRITE</Permission>
</Grant>

<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
    <ID>user2-canonical-user-ID</ID>
    <DisplayName>display-name</DisplayName>
  </Grantee>
  <Permission>READ</Permission>
</Grant>

<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
  </Grantee>
  <Permission>READ</Permission>
</Grant>
<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
  </Grantee>
  <Permission>WRITE</Permission>
</Grant>

</AccessControlList>
</AccessControlPolicy>

```

In Dosen ACL

Amazon S3 unterstützt eine Reihe vordefinierter Zuschüsse, die als vorgefertigte Zuschüsse bezeichnet ACLs werden. Jeder Status ACL enthält einen vordefinierten Satz von Stipendiaten

und Berechtigungen. In der folgenden Tabelle sind die vordefinierten Zuschüsse ACLs und die zugehörigen vordefinierten Zuschüsse aufgeführt.

Eingesetzt ACL	Gilt für	Berechtigungen wurden hinzugefügt zu ACL
<code>private</code>	Bucket und Objekt	Der Eigentümer erhält <code>FULL_CONTROL</code> . Niemand anderer hat Zugriffsrechte (Standard).
<code>public-read</code>	Bucket und Objekt	Der Eigentümer erhält <code>FULL_CONTROL</code> . Die <code>AllUsers</code> -Gruppe (siehe Wer ist ein Empfänger?) erhält <code>READ</code> -Zugriff.
<code>public-read-write</code>	Bucket und Objekt	Der Eigentümer erhält <code>FULL_CONTROL</code> . Die <code>AllUsers</code> -Gruppe erhält <code>READ</code> - und <code>WRITE</code> -Zugriff. Eine solche Erteilung von Rechten für einen Bucket wird im Allgemeinen nicht empfohlen.
<code>aws-exec-read</code>	Bucket und Objekt	Der Eigentümer erhält <code>FULL_CONTROL</code> . Amazon EC2 erhält <code>READ</code> Zugriff auf <code>GET</code> ein Amazon Machine Image (AMI) -Paket von Amazon S3.
<code>authenticated-read</code>	Bucket und Objekt	Der Eigentümer erhält <code>FULL_CONTROL</code> . Die <code>AuthenticatedUsers</code> -Gruppe erhält <code>READ</code> -Zugriff.
<code>bucket-owner-read</code>	Objekt	Der Objekt-Eigentümer erhält <code>FULL_CONTROL</code> . Der Bucket-Eigentümer erhält <code>READ</code> -Zugriff. Wenn Sie ACL bei der Erstellung eines Buckets angeben, dass dieser Wert gespeichert ist, ignoriert Amazon S3 ihn.
<code>bucket-owner-full-control</code>	Object	Sowohl der Objekt-Eigentümer, als auch der Bucket-Eigentümer erhalten <code>FULL_CONTROL</code> für das Objekt. Wenn Sie ACL bei der Erstellung eines Buckets angeben, dass dieser Wert gespeichert ist, ignoriert Amazon S3 ihn.
<code>log-delivery-write</code>	Bucket	Die <code>LogDelivery</code> -Gruppe erhält <code>WRITE</code> - und <code>READ_ACP</code> -Berechtigungen für den Bucket. Weitere

Eingesetzt ACL	Gilt für	Berechtigungen wurden hinzugefügt zu ACL
		Informationen über Protokolle finden Sie unter (Protokollieren von Anfragen mit Server-Zugriffsprotokollierung).

Note

Sie können ACLs in Ihrer Anfrage nur eines dieser vorgefertigten Objekte angeben.

Mithilfe des Anforderungsheaders geben Sie ACL in Ihrer Anfrage einen `x-amz-ac1` vorgefertigten Wert an. Wenn Amazon S3 eine Anfrage mit einer gespeicherten ACL Anfrage erhält, werden die vordefinierten Zuschüsse ACL der Ressource hinzugefügt.

Konfiguration ACLs

In diesem Abschnitt wird erklärt, wie Sie Zugriffsberechtigungen für S3-Buckets und -Objekte mithilfe von Zugriffskontrolllisten (ACLs) verwalten. Sie können Ihrer Ressource ACL mithilfe der AWS Management Console, AWS Command Line Interface (CLI), RESTAPI, oder AWS SDKs.

Bucket- und Objekt-Berechtigungen sind voneinander unabhängig. Ein Objekt erbt nicht die Berechtigungen von seinem Bucket. Wenn Sie beispielsweise einen Bucket erstellen und einem Benutzer Schreibzugriff erteilen, können Sie auf die Objekte dieses Benutzers nicht zugreifen, wenn Ihnen der Benutzer nicht explizit Zugriff erteilt.

Sie können anderen Berechtigungen gewähren AWS-Konto Benutzern oder vordefinierten Gruppen. Der Benutzer oder die Gruppe, dem bzw. der Sie Berechtigungen erteilen, wird als der Berechtigungsempfänger bezeichnet. Standardmäßig der Besitzer, das ist AWS-Konto der den Bucket erstellt hat, hat volle Berechtigungen.

Jede Berechtigung, die Sie einem Benutzer oder einer Gruppe gewähren, fügt einen Eintrag in dem `hinzACL`, der dem Bucket zugeordnet ist. ACLIn der Liste werden Zuschüsse aufgeführt, anhand derer der Empfänger und die erteilte Berechtigung identifiziert werden.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs

diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner enforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Warning

Es wird dringend empfohlen, der Gruppe Jeder (öffentlicher Zugriff) oder der Gruppe Authentifizierte Benutzer (alle) keinen Schreibzugriff zu gewähren AWS Gruppen (authentifizierte Benutzer). Weitere Informationen zu den Auswirkungen eines Schreibzugriffs für diese Gruppen finden Sie unter [Vordefinierte Gruppen in Amazon S3](#).

Verwenden der S3-Konsole zum Festlegen von ACL Berechtigungen für einen Bucket

Die Konsole zeigt kombinierte Zugriffsberechtigungen für doppelte Berechtigungsempfänger an. Um die vollständige Liste von zu sehen ACLs, verwenden Sie Amazon S3 RESTAPI, AWS CLI, oder AWS SDKs.

Die folgende Tabelle zeigt die ACL Berechtigungen, die Sie für Buckets in der Amazon S3 S3-Konsole konfigurieren können.

Amazon S3 ACL S3-Konsolenberechtigungen für Buckets

Konsolenberechtigung	ACLErlaubnis	Zugriff
Objekte – Auflisten	READ	Gestattet dem Empfänger, die Objekte im Bucket aufzulisten.
Objekte – Schreiben	WRITE	Gestattet dem Empfänger, neue Objekte im Bucket zu erstellen . Für die Bucket- und Objekteigentümer vorhandener Objekte können auch Löschungen und Überschreibungen dieser Objekte ermöglicht werden.
Eimer ACL — Lesen	READ_ACP	Ermöglicht dem Empfänger, den Bucket ACL zu lesen.
Eimer ACL — Schreiben	WRITE_ACP	Ermöglicht dem Empfänger, das ACL für den entsprechenden Bucket zu schreiben.
Jeder (öffentlicher Zugang): Objekte – Auflisten	READ	Gewährt öffentlichen Lesezugriff für die Objekte im Bucket. Wenn Sie jedem (öffentlichen Zugriff) Listenzugriff gewähren, kann jeder Benutzer auf der Welt auf die Objekte im Bucket zugreifen.
Jeder (öffentlich zugänglich): Bucket ACL — Lesen	READ_ACP	Gewährt öffentlichen Lesezugriff für den BucketACL. Wenn Sie allen Benutzern Lesezugriff gewähren (öffentlichen Zugriff), kann jeder auf der Welt auf den Bucket zugreifenACL.

Weitere Informationen zu ACL Berechtigungen finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

⚠ Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner enforced

aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Um ACL Berechtigungen für einen Bucket festzulegen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Berechtigungen festlegen möchten.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Access Control List Bearbeiten.

Sie können die folgenden ACL Berechtigungen für den Bucket bearbeiten:

Objekte

- Auflisten – Gestattet einem Empfänger, die Objekte im Bucket aufzulisten.
- Schreiben — Ermöglicht es dem Berechtigungsempfänger, neue Objekte im Bucket zu erstellen. Für die Bucket- und Objekteigentümer vorhandener Objekte können auch Löschungen und Überschreibungen dieser Objekte ermöglicht werden.

In der S3-Konsole können Sie nur der S3-Protokollbereitstellungsgruppe und dem Bucket-Besitzer (Ihr AWS-Konto). Wir empfehlen dringend, anderen Stipendiaten keinen Schreibzugriff zu gewähren. Wenn Sie jedoch Schreibzugriff gewähren müssen, können Sie den AWS CLI, AWS SDKs, oder das RESTAPI.


Eimer ACL

- Lesen — Ermöglicht dem Empfänger, den Bucket ACL zu lesen.
 - Schreiben — Ermöglicht dem Empfänger, das ACL für den entsprechenden Bucket zu schreiben.
5. Um die Berechtigungen des Bucket-Besitzers zu ändern, klicken Sie neben dem Bucket-Besitzer (Ihr AWS-Konto), deaktivieren Sie die folgenden ACL Berechtigungen oder wählen Sie eine aus:
 - Objekte – Auflisten oder Schreiben

- Bucket ACL — Lesen oder Schreiben

Der Besitzer bezieht sich auf Root-Benutzer des AWS-Kontos, kein AWS Identity and Access Management IAMNutzer. Weitere Informationen über den Root-Benutzer finden Sie [unter Root-Benutzer des AWS-Kontos](#) im IAM-Benutzerhandbuch.

6. Wenn Sie Berechtigungen für die allgemeine Öffentlichkeit (jeder im Internet) gewähren oder rückgängig machen möchten, deaktivieren Sie neben Jeder (öffentlicher Zugriff) die folgenden Berechtigungen oder wählen Sie eine der folgenden ACL Berechtigungen aus:
 - Objekte – Auflisten
 - Bucket ACL — Lesen

 Warning


Seien Sie vorsichtig, wenn Sie der Gruppe Everyone (Jeder) öffentlichen Zugriff auf Ihren S3-Bucket gewähren. Wenn Sie dieser Gruppe anonymen Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihren Bucket zugreifen. Wir empfehlen dringend, nie einen öffentlichen Schreibzugriff auf Ihren S3-Bucket zu gewähren.

7. Um Berechtigungen für eine Person zu gewähren oder rückgängig zu machen AWS-Konto, neben der Gruppe Authentifizierte Benutzer (jeder mit einem AWS-Konto), deaktivieren Sie die folgenden ACL Berechtigungen oder wählen Sie eine aus:
 - Objekte – Auflisten
 - Bucket ACL — Lesen
8. Um Amazon S3 Berechtigungen zum Schreiben von Serverzugriffsprotokollen in den Bucket zu gewähren oder rückgängig zu machen, deaktivieren Sie unter S3-Protokollbereitstellungsgruppe die folgenden ACL Berechtigungen oder wählen Sie eine aus:
 - Objekte – Auflisten oder Schreiben
 - Bucket ACL — Lesen oder Schreiben

Wenn ein Bucket als Ziel-Bucket für Zugriffsprotokolle eingerichtet ist, müssen die Bucket-Berechtigungen der Gruppe Log Delivery (Protokollbereitstellung) Schreibzugriff auf den Bucket erteilen. Wenn Sie die Server-Zugriffsprotokollierung für einen Bucket aktivieren, erteilt die Amazon-S3-Konsole der Gruppe Log Delivery (Protokollbereitstellung) Schreibzugriff

auf den Ziel-Bucket, den Sie für den Empfang der Protokolle ausgewählt haben. Weitere Informationen zu Server-Zugriffsprotokollen finden Sie unter [Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung](#).

9. Um einem anderen Zugriff zu gewähren AWS-Konto wie folgt:
 - a. Wählen Sie Empfänger hinzufügen.
 - b. Geben Sie in das Feld Stipendiat die kanonische ID des anderen ein AWS-Konto.
 - c. Wählen Sie aus den folgenden Berechtigungen aus: ACL
 - Objekte – Auflisten oder Schreiben
 - Bucket ACL — Lesen oder Schreiben

 Warning

Wenn Sie andere gewähren AWS-Konten Zugriff auf Ihre Ressourcen, beachten Sie, dass AWS-Konten können ihre Berechtigungen an Benutzer unter ihren Konten delegieren. Man spricht auch von einem kontenübergreifenden Zugriff. Informationen zur Verwendung des kontoübergreifenden Zugriffs finden Sie unter [Erstellen einer Rolle zur Delegierung von Berechtigungen an einen IAM Benutzer im IAM Benutzerhandbuch](#).

10. So entziehen Sie einem anderen Benutzer den Zugriff AWS-Konto, unter Zugriff für andere AWS-Konten, wählen Sie Entfernen.
11. Klicken Sie auf Save changes (Änderungen speichern), um die Änderungen zu speichern.

Verwenden Sie die S3-Konsole, um ACL Berechtigungen für ein Objekt festzulegen

Die Konsole zeigt kombinierte Zugriffsberechtigungen für doppelte Berechtigungsempfänger an. Um die vollständige Liste von zu sehen ACLs, verwenden Sie Amazon S3 RESTAPI, AWS CLI, oder AWS SDKs. Die folgende Tabelle zeigt die ACL Berechtigungen, die Sie für Objekte in der Amazon S3 S3-Konsole konfigurieren können.

Amazon S3 ACL S3-Konsolenberechtigungen für Objekte

Konsolenberechtigung	ACLErlaubnis	Zugriff
Objekt – Lesen	READ	Gestattet dem Empfänger, die Objektdaten und seine Metadaten zu lesen.
Objekt ACL — Lesen	READ_ACP	Ermöglicht dem Empfänger, das Objekt ACL zu lesen.
Objekt ACL — Schreiben	WRITE_ACP	Ermöglicht dem Empfänger, das ACL für das entsprechende Objekt zu schreiben

Weitere Hinweise zu ACL Berechtigungen finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner enforced aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den AccessControlListNotSupported Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Um ACL Berechtigungen für ein Objekt festzulegen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objekte den Namen des Objekts aus, für das Sie Berechtigungen festlegen möchten.
4. Wählen Sie Permissions (Berechtigungen).
5. Wählen Sie unter Zugriffskontrollliste (ACL) die Option Bearbeiten aus.

Sie können die folgenden ACL Berechtigungen für das Objekt bearbeiten:

Object

- Lesen – Gestattet dem Empfänger, die Objektedaten und seine Metadaten zu lesen.

Objekt ACL

- Lesen — Ermöglicht dem Empfänger, das Objekt ACL zu lesen.
- Schreiben — Ermöglicht dem Empfänger, das ACL für das entsprechende Objekt zu schreiben. In der S3-Konsole können Sie nur dem Bucket-Besitzer Schreibzugriff gewähren (Ihr AWS-Konto). Wir empfehlen dringend, anderen Stipendiaten keinen Schreibzugriff zu gewähren. Wenn Sie jedoch Schreibzugriff gewähren müssen, können Sie den AWS CLI, AWS SDKs, oder das RESTAPI.

6. Sie können Zugriffsberechtigungen von Objekten für Folgendes verwalten:

a. Zugriff für den Besitzer des Objekts

Der Besitzer bezieht sich auf den Root-Benutzer des AWS-Kontos, und kein AWS Identity and Access Management IAM-Benutzer. Weitere Informationen über den Root-Benutzer finden Sie [unter Root-Benutzer des AWS-Kontos](#) im IAM-Benutzerhandbuch.

Um die Objektzugriffsberechtigungen des Besitzers zu ändern, wählen Sie unter Zugriff für Objekteigentümer die Option Ihr AWS Konto (Besitzer).

Aktivieren Sie die Kontrollkästchen für die Berechtigungen, die geändert werden sollen. Wählen Sie dann Save (Speichern) aus.

b. Zugang für andere AWS-Konten

Um einem Benutzer Berechtigungen zu erteilen AWS Benutzer aus einem anderen AWS-Konto, unter Zugriff für andere AWS-Konten, wählen Sie Konto hinzufügen. Geben Sie im Feld Geben Sie eine ID ein die kanonische ID des AWS Benutzer, dem Sie Objektberechtigungen erteilen möchten. Informationen zur Suche nach einer kanonischen ID finden Sie unter Ihr [AWS-Konto Identifikatoren](#) in der Allgemeine Amazon Web Services-Referenz. Sie können bis zu 99 Benutzer hinzufügen.

Aktivieren Sie die Kontrollkästchen für die Berechtigungen, die dem Benutzer gewährt werden sollen. Wählen Sie dann Save (Speichern) aus. Um Informationen über die Berechtigungen anzuzeigen, wählen Sie die Hilfesymbole aus.

c. Öffentlicher Zugriff

Um der allgemeinen Öffentlichkeit (jedem Benutzer auf der ganzen Welt) Zugriff auf Ihr Objekt zu gewähren, wählen Sie unter Public access (Öffentlicher Zugriff) Everyone (Jeder) aus. Die Erteilung öffentlicher Zugriffsberechtigungen bedeutet, dass jeder Benutzer auf der ganzen Welt auf das Objekt zugreifen kann.

Aktivieren Sie die Kontrollkästchen für die Berechtigungen, die erteilt werden sollen. Wählen Sie dann Save (Speichern) aus.

Warning

- Seien Sie vorsichtig, wenn Sie der Gruppe Everyone (Jeder) anonymen Zugriff auf Ihre Amazon-S3-Objekte gewähren. Wenn Sie dieser Gruppe Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihr Objekt zugreifen. Wenn Sie jedem Zugriff gewähren müssen, wird nachdrücklich empfohlen, dass Sie nur Berechtigungen für Read objects (Objekte lesen) gewähren.
- Wir empfehlen nachdrücklich, der Gruppe Everyone (Jeder) keine Schreibberechtigungen für Objekte zu erteilen. Auf diese Weise kann jeder die ACL Berechtigungen für das Objekt überschreiben.

Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele für die Konfiguration von Zugriffssteuerungslisten (ACL) für Buckets und Objekte.

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner enforced aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Java

Dieser Abschnitt enthält Beispiele für die Konfiguration von Zugriffssteuerungslisten (ACL) für Buckets und Objekte. Im ersten Beispiel wird ein Bucket mit einem vordefinierten Wert erstellt ACL (siehe [In Dosen ACL](#)), eine Liste mit benutzerdefinierten Berechtigungen erstellt und anschließend das Bucket durch ein Bucket ACL ersetzt, das die benutzerdefinierten Berechtigungen ACL enthält. Das zweite Beispiel zeigt, wie die `AccessControlList.grantPermission()` Methode geändert und ACL verwendet wird.

Example Erstellen Sie einen Bucket und geben Sie einen Bucket an ACL, der der S3-Protokollzustellungsgruppe die entsprechenden Berechtigungen erteilt

Dieses Beispiel erstellt einen Bucket. In der Anfrage gibt das Beispiel einen Container an, der der Log Delivery-Gruppe die Berechtigung erteilt, Logs in den Bucket zu schreiben. ACL

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.ArrayList;

public class CreateBucketWithACL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String userEmailForReadPermission = "*** user@example.com ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .build();

            // Create a bucket with a canned ACL. This ACL will be replaced by the
            // setBucketAcl()
            // calls below. It is included here for demonstration purposes.
            CreateBucketRequest createBucketRequest = new
            CreateBucketRequest(bucketName, clientRegion.getName())
```

```
        .withCannedAcl(CannedAccessControlList.LogDeliveryWrite);
    s3Client.createBucket(createBucketRequest);

    // Create a collection of grants to add to the bucket.
    ArrayList<Grant> grantCollection = new ArrayList<Grant>();

    // Grant the account owner full control.
    Grant grant1 = new Grant(new
CanonicalGrantee(s3Client.getS3AccountOwner().getId()),
        Permission.FullControl);
    grantCollection.add(grant1);

    // Grant the LogDelivery group permission to write to the bucket.
    Grant grant2 = new Grant(GroupGrantee.LogDelivery, Permission.Write);
    grantCollection.add(grant2);

    // Save grants by replacing all current ACL grants with the two we just
created.
    AccessControlList bucketAcl = new AccessControlList();
    bucketAcl.grantAllPermissions(grantCollection.toArray(new Grant[0]));
    s3Client.setBucketAcl(bucketName, bucketAcl);

    // Retrieve the bucket's ACL, add another grant, and then save the new
ACL.
    AccessControlList newBucketAcl = s3Client.getBucketAcl(bucketName);
    Grant grant3 = new Grant(new
EmailAddressGrantee(userEmailForReadPermission), Permission.Read);
    newBucketAcl.grantAllPermissions(grant3);
    s3Client.setBucketAcl(bucketName, newBucketAcl);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
// it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

Example Aktualisieren ACL Sie ein vorhandenes Objekt

In diesem Beispiel wird das ACL für ein Objekt aktualisiert. Das Beispiel führt die folgenden Aufgaben durch:

- Ruft die eines Objekts ab ACL
- Löscht die, ACL indem alle vorhandenen Berechtigungen entfernt werden
- Fügt zwei Berechtigungen hinzu: Vollzugriff für den Besitzer und WRITE _ ACP (siehe [Welche Berechtigungen kann ich erteilen?](#)) für einen Benutzer, der durch eine E-Mail-Adresse identifiziert wird
- Speichert ACL das im Objekt

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AccessControlList;
import com.amazonaws.services.s3.model.CanonicalGrantee;
import com.amazonaws.services.s3.model.EmailAddressGrantee;
import com.amazonaws.services.s3.model.Permission;

import java.io.IOException;

public class ModifyACLExistingObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Key name ****";
        String emailGrantee = "**** user@example.com ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Get the existing object ACL that we want to modify.
```



```
        AccessControlList acl = s3Client.getObjectAcl(bucketName, keyName);

        // Clear the existing list of grants.
        acl.getGrantsAsList().clear();

        // Grant a sample set of permissions, using the existing ACL owner for
Full
        // Control permissions.
        acl.grantPermission(new CanonicalGrantee(acl.getOwner().getId()),
Permission.FullControl);
        acl.grantPermission(new EmailAddressGrantee(emailGrantee),
Permission.WriteAcp);

        // Save the modified ACL back to the object.
        s3Client.setObjectAcl(bucketName, keyName, acl);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Example Erstellen Sie einen Bucket und geben Sie einen Bucket anACL, der der S3-Protokollzustellungsgruppe die entsprechenden Berechtigungen erteilt

Dieses C#-Beispiel erstellt einen Bucket. In der Anfrage gibt der Code auch einen Container an, der der Log Delivery-Gruppe die Berechtigung erteilt, die Logs in den Bucket zu schreiben. ACL

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;
```

```
namespace Amazon.DocSamples.S3
{
    class ManagingBucketACLTest
    {
        private const string newBucketName = "**** bucket name ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateBucketUseCannedACLAsync().Wait();
        }

        private static async Task CreateBucketUseCannedACLAsync()
        {
            try
            {
                // Add bucket (specify canned ACL).
                PutBucketRequest putBucketRequest = new PutBucketRequest()
                {
                    BucketName = newBucketName,
                    BucketRegion = S3Region.EUW1, // S3Region.US,
                                                    // Add canned ACL.
                    CannedACL = S3CannedACL.LogDeliveryWrite
                };
                PutBucketResponse putBucketResponse = await
client.PutBucketAsync(putBucketRequest);

                // Retrieve bucket ACL.
                GetACLResponse getACLResponse = await client.GetACLAsync(new
GetACLRequest
                {
                    BucketName = newBucketName
                });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("S3 error occurred. Exception: " +
amazonS3Exception.ToString());
            }
        }
    }
}
```

```
        catch (Exception e)
        {
            Console.WriteLine("Exception: " + e.ToString());
        }
    }
}
```

Example Aktualisieren ACL Sie ein vorhandenes Objekt

In diesem C#-Beispiel wird das ACL für ein vorhandenes Objekt aktualisiert. Das Beispiel führt die folgenden Aufgaben durch:

- Ruft die eines Objekts ab. ACL
- Löscht die, ACL indem alle vorhandenen Berechtigungen entfernt werden.
- Fügt zwei Berechtigungen hinzu: Vollzugriff für den Besitzer und WRITE _ ACP für einen Benutzer, der anhand einer E-Mail-Adresse identifiziert wird.
- Speichert das, ACL indem es eine PutAc1 Anfrage sendet.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingObjectACLTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string keyName = "**** object key name ****";
        private const string emailAddress = "**** email address ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
```

```
{
    client = new AmazonS3Client(bucketRegion);
    TestObjectACLTestAsync().Wait();
}
private static async Task TestObjectACLTestAsync()
{
    try
    {
        // Retrieve the ACL for the object.
        GetACLResponse aclResponse = await client.GetACLAsync(new
GetACLRequest
        {
            BucketName = bucketName,
            Key = keyName
        });

        S3AccessControlList acl = aclResponse.AccessControlList;

        // Retrieve the owner (we use this to re-add permissions after
we clear the ACL).
        Owner owner = acl.Owner;

        // Clear existing grants.
        acl.Grants.Clear();

        // Add a grant to reset the owner's full permission (the
previous clear statement removed all permissions).
        S3Grant fullControlGrant = new S3Grant
        {
            Grantee = new S3Grantee { CanonicalUser = owner.Id },
            Permission = S3Permission.FULL_CONTROL
        };

        // Describe the grant for the permission using an email address.
        S3Grant grantUsingEmail = new S3Grant
        {
            Grantee = new S3Grantee { EmailAddress = emailAddress },
            Permission = S3Permission.WRITE_ACP
        };
        acl.Grants.AddRange(new List<S3Grant> { fullControlGrant,
grantUsingEmail });

        // Set a new ACL.
```

```
        PutACLResponse response = await client.PutACLAsync(new
PutACLRequest
    {
        BucketName = bucketName,
        Key = keyName,
        AccessControlList = acl
    });
    }
    catch (AmazonS3Exception amazonS3Exception)
    {
        Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
amazonS3Exception.ToString());
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception: " + e.ToString());
    }
    }
}
}
```

Mit dem REST API

Mit Amazon APIs S3 können Sie festlegen, ACL wann Sie einen Bucket oder ein Objekt erstellen. Amazon S3 bietet auch API die Möglichkeit, einen ACL auf einen vorhandenen Bucket oder ein Objekt zu setzen. Diese APIs bieten die folgenden Methoden zum Setzen eines ACL:

- ACL Mithilfe von Anforderungsheadern festlegen— Wenn Sie eine Anfrage zum Erstellen einer Ressource (Bucket oder Objekt) senden, legen Sie eine ACL mithilfe der Anforderungsheader fest. Mit diesen Headern können Sie entweder einen festgelegten Wert angeben ACL oder Grants explizit angeben (wobei der Empfänger und die Berechtigungen explizit identifiziert werden).
- ACL Mithilfe des Anfragetextes festlegen — Wenn Sie eine Anfrage zum Einstellen eines für eine ACL vorhandene Ressource senden, können Sie dies ACL entweder im Anforderungsheader oder im Hauptteil der Anfrage festlegen.

Informationen zur REST API Unterstützung bei der Verwaltung ACLs finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [GETBucket-ACL](#)

- [PUTEimer ACL](#)
- [GETObjekt ACL](#)
- [PUTObjekt ACL](#)
- [PUTObjekt](#)
- [PUTEimer](#)
- [PUTObjekt — Kopieren](#)
- [Initiieren eines mehrteiligen Uploads](#)

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Besitzer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien für die Gewährung des Zugriffs auf Ihren Bucket und die enthaltenen Objekte verwenden. Wenn die Einstellung Bucket Owner enforced aktiviert ist, schlagen Anfragen zum Einrichten von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Spezifische Anforderungsheader für Access Control List (ACL)

Sie können Header verwenden, um auf Zugriffskontrolllisten (ACL) basierende Berechtigungen zu erteilen. Standardmäßig sind alle Objekte privat. Nur der Besitzer hat die vollständige Zugriffssteuerung. Wenn Sie ein neues Objekt hinzufügen, können Sie einzelnen Personen Berechtigungen gewähren AWS-Konten oder zu vordefinierten Gruppen, die von Amazon S3 definiert wurden. Diese Berechtigungen werden dann der Zugriffskontrollliste (ACL) für das Objekt hinzugefügt. Weitere Informationen finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Mit dieser Operation können Sie Zugriffsberechtigungen mit einer der folgenden beiden Methoden erteilen:

- Canned ACL (**x-amz-acl**) — Amazon S3 unterstützt eine Reihe von vordefinierten ACLs, sogenannten Canned ACLs. Jeder Scan ACL hat einen vordefinierten Satz von Empfängern und Berechtigungen. Weitere Informationen finden Sie unter [In Dosen ACL](#).
- Zugriffsberechtigungen — Um bestimmten Personen explizit Zugriffsberechtigungen zu gewähren AWS-Konten verwenden Sie für Gruppen die folgenden Header. Jeder Header ist bestimmten Berechtigungen zugeordnet, die Amazon S3 in einem unterstützt ACL. Weitere Informationen

finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#). Im Header geben Sie eine Liste der Empfänger an, die die jeweilige Berechtigung erhalten.

- x-amz-grant-read
- x-amz-grant-write
- x-amz-grant-read-acp
- x-amz-grant-write-acp
- x-amz-grant-full-kontrollieren

Verwendung der AWS CLI

Weitere Informationen zur Verwaltung ACLs mit dem AWS CLI, siehe [put-bucket-acl](#) in der AWS CLI Befehlsreferenz.

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner enforced aktiviert ist, schlagen Anfragen zum Einrichten von Zugriffskontrolllisten (ACLs) oder zum Aktualisieren ACLs fehl und geben den AccessControlListNotSupported Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Beispiele für Richtlinien für ACLs

Sie können Bedingungsschlüssel in Bucket-Richtlinien verwenden, um den Zugriff auf Amazon S3 zu kontrollieren.

Themen

- [Gewährung von PutObject s3-Berechtigungen mit einer Bedingung, die voraussetzt, dass der Bucket-Besitzer die volle Kontrolle erhält](#)
- [Erteilung der PutObject s3-Berechtigung mit einer Bedingung im x-amz-acl Header](#)

Gewährung von PutObject s3-Berechtigungen mit einer Bedingung, die voraussetzt, dass der Bucket-Besitzer die volle Kontrolle erhält

Die Operation [PUTObjekt](#) ermöglicht eine Zugriffskontrollliste (ACL), d. h. Header, die Sie verwenden können, um bestimmte Berechtigungen zu erteilenACL. Mit diesen Schlüsseln kann der Bucket-Eigentümer eine Bedingung festlegen, die bestimmte Zugriffsberechtigungen erfordert, wenn der Benutzer ein Objekt hochlädt.

Angenommen, Konto A besitzt einen Bucket und der Kontoadministrator möchte Dave, einem Benutzer in Konto B, Berechtigungen zum Hochladen von Objekten erteilen. Standardmäßig gehören Objekte, die Dave hochgeladen hat, zu Konto B, und Konto A besitzt keine Berechtigungen für diese Objekte. Da der Bucket-Eigentümer die Rechnungen bezahlt, benötigt er volle Berechtigungen für die Objekte, die Dave hochlädt. Das Konto Ein Administrator kann dies tun, indem er Dave die s3:PutObject entsprechende Berechtigung erteilt, allerdings mit der Bedingung, dass die Anfrage ACL spezifische Header enthält, die entweder explizit volle Rechte gewähren oder vordefinierte Header verwenden. ACL [Weitere Informationen finden Sie unter PUT Objekt](#).

Erfordert den x-amz-full-control Header

Sie können in der Anforderung den Header x-amz-full-control mit vollständiger Kontrollberechtigung für den Bucket-Eigentümer anfordern. Die folgende Bucket-Richtlinie erteilt dem Benutzer Dave die Berechtigung s3:PutObject mit einer Bedingung, die den Bedingungsschlüssel s3:x-amz-grant-full-control enthält, in die Anforderung den Header x-amz-full-control aufzunehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:user/Dave"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
        }
      }
    }
  ]
}
```



```

    }
  ]
}
```

Note

In diesem Beispiel geht es um die kontoübergreifende Berechtigung. Wenn Dave (der die Erlaubnis erhält) jedoch zu den AWS-Konto dem der Bucket gehört, ist diese bedingte Erlaubnis nicht erforderlich. Grund hierfür ist, dass das übergeordnete Konto, dem Dave angehört, Eigentümer von Objekten ist, die der Benutzer hochlädt.

Hinzufügen einer expliziten Zugriffsverweigerung

Die vorherige Bucket-Richtlinie erteilt dem Benutzer Dave in Konto B eine bedingte Berechtigung. Während diese Richtlinie in Kraft ist, ist es Dave möglich, die gleiche Berechtigung ohne Bedingung über eine andere Richtlinie zu erhalten. Beispiel: Dave kann einer Gruppe angehören, der Sie die Berechtigung `s3:PutObject` bedingungslos erteilen. Um solche Berechtigungslücken zu vermeiden, können Sie eine strengere Zugriffsrichtlinie schreiben, indem Sie eine explizite Zugriffsverweigerung hinzufügen. In diesem Beispiel verweigern Sie explizit die Upload-Berechtigung für Benutzer Dave, wenn er nicht die erforderlichen Header in die Anforderung einbindet, die dem Bucket-Eigentümer volle Berechtigungen gewährt. Eine explizite Verweigerung ersetzt immer eine anderswo erteilte Erlaubnis. Im Folgenden finden Sie das geänderte Zugriffsrichtlinienbeispiel mit hinzugefügter expliziter Ablehnung.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:user/AccountAdmin"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::awsexamplebucket1/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Sid": "statement2",
    "Effect": "Deny",
    "Principal": {
      "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3::awsexamplebucket1/*",
    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
      }
    }
  }
]
}

```

Testen Sie die Richtlinie mit AWS CLI

Wenn Sie zwei haben AWS-Konten, können Sie die Richtlinie mit dem testen AWS Command Line Interface (AWS CLI). Sie hängen die Richtlinie an und verwenden Daves Anmeldeinformationen, um die Berechtigung wie folgt zu testen AWS CLI put-object-Befehl. Sie stellen dem Benutzer Dave die Anmeldeinformationen bereit, indem Sie den Parameter `--profile` hinzufügen. Sie erteilen dem Bucket-Eigentümer die volle Kontrolle, indem Sie den Parameter `--grant-full-control` hinzufügen. Weitere Informationen zur Einrichtung und Verwendung des AWS CLI, finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
--grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

Erfordert den x-amz-acl Header

Sie können den `x-amz-acl` Header mit einem voreingestellten Header anfordern, der dem Bucket-Besitzer volle Zugriffsrechte ACL gewährt. Um den Header `x-amz-acl` in der Anforderung erforderlich zu machen, können Sie das Schlüssel-Wert-Paar im Block `Condition` ersetzen und den Bedingungsschlüssel `s3:x-amz-acl` wie im folgenden Beispiel dargestellt angeben.

```
"Condition": {
  "StringEquals": {
```

```

    "s3:x-amz-acl": "bucket-owner-full-control"
  }
}

```

Um die Berechtigung zu testen, verwenden Sie AWS CLI, geben Sie den `--acl` Parameter an. Das Tool AWS CLI fügt dann den `x-amz-acl` Header hinzu, wenn die Anfrage gesendet wird.

```

aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
--acl "bucket-owner-full-control" --profile AccountBadmin

```

Erteilung der PutObject s3-Berechtigung mit einer Bedingung im x-amz-acl Header

Die folgende Bucket-Richtlinie gewährt die `s3:PutObject` Erlaubnis für zwei AWS-Konten wenn die Anfrage den `x-amz-acl` Header enthält, der das Objekt öffentlich lesbar macht. Der Condition-Block verwendet die `StringEquals`-Bedingung und ist mit dem Schlüssel-Wert-Paar `"s3:x-amz-acl":["public-read"]` zur Auswertung versehen. Im Schlüssel-Wert-Paar ist `s3:x-amz-acl` ein Amazon S3-spezifischer Schlüssel, wie durch das Präfix `s3:` angezeigt ist.

```

{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AddCannedAcl",
      "Effect":"Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::Account1-ID:root",
          "arn:aws:iam::Account2-ID:root"
        ]
      },
      "Action":"s3:PutObject",
      "Resource": ["arn:aws:s3::awsexamplebucket1/*"],
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl":["public-read"]
        }
      }
    }
  ]
}

```

⚠ Important

Nicht alle Bedingungen machen für alle Aktionen auch Sinn. Beispielsweise ist es sinnvoll, die Bedingung `s3:LocationConstraint` für eine Richtlinie einzufügen, die die Amazon-S3-Berechtigung `s3:CreateBucket` erteilt. Es ist jedoch nicht sinnvoll, diese Bedingung in eine Richtlinie aufzunehmen, die die `s3:GetObject`-Genehmigung erteilt. Amazon S3 kann auf semantische Fehler dieses Typs testen, die Amazon-S3-spezifische Bedingungen beinhalten. Wenn Sie jedoch eine Richtlinie für einen IAM Benutzer oder eine Rolle erstellen und eine semantisch ungültige Amazon S3 S3-Bedingung angeben, wird kein Fehler gemeldet, da die Amazon S3 S3-Bedingungen IAM nicht validiert werden können.

Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher

Die Amazon S3 Block Public Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff. Benutzer können jedoch Bucket-Richtlinien, Zugriffspunkt-Richtlinien oder Objektberechtigungen ändern, um öffentlichen Zugriff zu ermöglichen. S3 Block Public Access-Einstellungen überschreiben diese Richtlinien und Berechtigungen, damit Sie den öffentlichen Zugriff auf diese Ressourcen einschränken können.

Mit S3 Block Public Access können Administratoren und Bucket-Eigentümer problemlos zentrale Kontrollen zur Begrenzung des öffentlichen Zugriffs auf ihre Amazon-S3-Ressourcen einrichten. Diese Kontrollen werden unabhängig davon durchgesetzt, wie die Ressourcen erstellt wurden.

Anweisungen zur Konfiguration von Public Block Access finden Sie unter [Verwenden von Block Public Access](#).

Wenn Amazon S3 eine Anforderung zum Zugriff auf einen Bucket oder ein Objekt erhält, wird ermittelt, ob für den Bucket oder das Konto des Bucket-Eigentümers eine Block Public Access-Einstellung vorliegt. Wenn die Anforderung über einen Zugriffspunkt einging, prüft Amazon S3 auch auf Block Public Access-Einstellungen für den Zugriffspunkt. Wenn eine Block Public Access-Einstellung vorhanden ist, die den angeforderten Zugriff verbietet, lehnt Amazon S3 die Anforderung ab.

Amazon S3 Block Public Access bietet vier Einstellungen. Diese Einstellungen sind voneinander unabhängig und können in beliebiger Kombination verwendet werden. Jede Einstellung kann auf

einen Access Point, einen Bucket oder ein ganzes Objekt angewendet werden AWS-Konto. Wenn sich die Einstellungen zum Sperren des öffentlichen Zugriffs für den Access Point, den Bucket oder das Konto unterscheiden, wendet Amazon S3 die restriktivste Kombination aus Access Point-, Bucket- und Kontoeinstellungen an.

Wenn Amazon S3 ermittelt, ob eine Operation von einer Block Public Access-Einstellung untersagt wird, werden alle Anforderungen abgelehnt, die gegen eine Zugriffspunkt-, Bucket- oder Konto-Einstellung verstoßen.

Important

Öffentlicher Zugriff auf Buckets und Objekte wird über Zugriffskontrolllisten (ACLs), Zugriffspunktrichtlinien, Bucket-Richtlinien oder alles gewährt. Um sicherzustellen, dass der öffentliche Zugriff für alle Amazon-S3-Zugriffspunkte, -Buckets und -Objekte blockiert ist, empfehlen wir, alle vier Einstellungen zur Blockierung des öffentlichen Zugriffs für das Konto zu aktivieren. Diese Einstellungen blockieren den öffentlichen Zugriff für alle aktuellen und künftigen Buckets und Zugriffspunkte.

Bevor Sie diese Einstellungen anwenden, verifizieren Sie, dass Ihre Anwendungen ohne öffentlichen Zugriff korrekt funktionieren. Wenn ein bestimmter Umfang an öffentlichem Zugriff auf Ihre Buckets oder Objekte nötig ist, z. B. zum Hosten einer statischen Website, wie unter [Hosten einer statischen Website mit Amazon S3](#) beschrieben, können Sie die einzelnen Einstellungen an Ihre Speicheranwendungsfälle anpassen.

Die Aktivierung von Block Public Access trägt zum Schutz Ihrer Ressourcen bei, indem verhindert wird, dass öffentlicher Zugriff über die Ressourcenrichtlinien oder Zugriffskontrolllisten (ACLs) gewährt wird, die direkt mit S3-Ressourcen verknüpft sind. Überprüfen Sie zusätzlich zur Aktivierung von Block Public Access sorgfältig die folgenden Richtlinien, um sicherzustellen, dass sie keinen öffentlichen Zugriff gewähren:

- Identitätspflichtige Richtlinien, die mit verbundenen verknüpft sind AWS Prinzipale (z. B. Rollen) IAM
- Ressourcenbasierte Richtlinien, die mit zugehörigen verknüpft sind AWS Ressourcen (zum Beispiel AWS Key Management Service (KMS) Schlüssel)

Note

- Sie können die Einstellungen zum Blockieren des öffentlichen Zugriffs nur für Access Points, Buckets und AWS-Konten. Amazon S3 unterstützt keine Einstellungen für den blockierten öffentlichen Zugriff auf Objektbasis.
- Wenn Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs auf ein Konto anwenden, gelten die Einstellungen für alle AWS-Regionen weltweit. Die Einstellungen werden möglicherweise nicht in allen Regionen umgehend oder gleichzeitig wirksam, werden aber auf jeden Fall von allen Regionen übernommen.

Themen


- [Block Public Access-Einstellungen](#)
- [Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt](#)
- [Die Bedeutung von „öffentlich“](#)
- [Verwenden von IAM Access Analyzer für S3 zur Überprüfung öffentlicher Buckets](#)
- [Berechtigungen](#)
- [Verwenden von Block Public Access](#)
- [Konfigurieren der Block-Public-Access-Einstellungen für Ihr Konto](#)
- [Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets](#)


Block Public Access-Einstellungen


S3 Block Public Access bietet vier Einstellungen. Sie können diese Einstellungen in beliebiger Kombination auf einzelne Access Points, Buckets oder ganze Access Points anwenden AWS-Konten. Wenn Sie eine Einstellung auf ein Konto anwenden, gilt sie für alle Buckets und Access Points, die diesem Konto gehören. Wenn Sie eine Einstellung auf einen Bucket anwenden, gilt diese auch für alle Zugriffspunkte, die diesem Bucket zugeordnet sind.

Die folgende Tabelle enthält die verfügbaren Einstellungen.

Name	Beschreibung
BlockPublicAccess	Das Festlegen dieser Option auf TRUE hat folgende Verhaltensweise zur Folge:

Name	Beschreibung
	<ul style="list-style-type: none">• PUTBucket-ACL- und PUT Object-ACL-Aufrufe schlagen fehl, wenn die angegebene Zugriffskontrollliste (ACL) öffentlich ist.• PUTObjektaufrufe schlagen fehl, wenn die Anfrage eine öffentliche Komponente enthältACL.• Wenn diese Einstellung auf ein Konto angewendet wird, schlagen PUT Bucket-Aufrufe fehl, wenn die Anfrage ein öffentliches Objekt beinhaltetACL. <p>Wenn diese Einstellung auf gesetzt istTRUE, schlagen die angegebenen Operationen fehl (unabhängig davon RESTAPI, ob sie über AWS CLI, oder AWS SDKs). Bestehende Richtlinien sowie Richtlinien ACLs für Buckets und Objekte werden jedoch nicht geändert. Mit dieser Einstellung können Sie sich vor öffentlichem Zugriff schützen und gleichzeitig die vorhandenen Richtlinien sowie Ihre Buckets und ACLs Objekte prüfen, verfeinern oder anderweitig ändern.</p> <div data-bbox="430 1087 1507 1545" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Access Points sind ihnen nicht ACLs zugeordnet. Wenn Sie diese Einstellung auf einen Zugriffspunkt anwenden, fungiert sie als Passthrough zum zugrunde liegenden Bucket. Wenn diese Einstellung für einen Zugriffspunkt aktiviert ist, verhalten sich Anforderungen, die über den Zugriffspunkt vorgenommen werden, so als ob der zugrunde liegende Bucket diese Einstellung aktiviert hat, unabhängig davon, ob diese Einstellung für den Bucket tatsächlich aktiviert ist.</p></div>

Name	Beschreibung
IgnorePublicAcls	<p>Wenn Sie diese Option TRUE ACLs auf setzen, ignoriert Amazon S3 alle öffentlichen Informationen in einem Bucket und alle darin enthaltenen Objekte. Diese Einstellung ermöglicht es Ihnen, öffentlichen Zugriff, der von gewährt wurde, sicher zu blockieren und ACLs gleichzeitig PUT Objekt-Aufrufe zuzulassen, die ein öffentliches Objekt enthalten ACL (im Gegensatz zu PUT ObjektaufrufenBlockPublicAcls , die ein öffentliches Objekt enthaltenACL). Die Aktivierung dieser Einstellung hat keinen Einfluss auf die Persistenz vorhandener Einstellungen ACLs und verhindert nicht, dass neue öffentliche Objekte eingerichtet ACLs werden.</p> <div data-bbox="428 684 1507 1140"><p> Note</p><p>Access Points sind ihnen nicht ACLs zugeordnet. Wenn Sie diese Einstellung auf einen Zugriffspunkt anwenden, fungiert sie als Passthrough zum zugrunde liegenden Bucket. Wenn diese Einstellung für einen Zugriffspunkt aktiviert ist, verhalten sich Anforderungen, die über den Zugriffspunkt vorgenommen werden, so als ob der zugrunde liegende Bucket diese Einstellung aktiviert hat, unabhängig davon, ob diese Einstellung für den Bucket tatsächlich aktiviert ist.</p></div>

Name	Beschreibung
BlockPublicPolicy	<p>Wenn Sie diese Option auf TRUE für einen Bucket setzen, lehnt Amazon S3 Aufrufe der PUT Bucket-Richtlinie ab, wenn die angegebene Bucket-Richtlinie öffentlichen Zugriff zulässt. Wenn diese Option auf TRUE für einen Bucket gesetzt wird, lehnt Amazon S3 auch Aufrufe der PUT Access Point-Richtlinie für alle Access Points des Buckets ab, die dasselbe Konto haben, wenn die angegebene Richtlinie öffentlichen Zugriff zulässt.</p> <p>Wenn Sie diese Option auf TRUE für einen Access Point setzen, lehnt Amazon S3 Aufrufe von PUT Access Point Policy und PUT Bucket Policy ab, die über den Access Point getätigt werden, wenn die angegebene Richtlinie (entweder für den Access Point oder den zugrunde liegenden Bucket) den öffentlichen Zugriff zulässt.</p> <p>Mit dieser Einstellung können Sie Benutzern erlauben, Zugriffspunkt- und Bucket-Richtlinien zu verwalten, ohne ihnen die öffentliche Freigabe des Buckets oder der darin enthaltenen Objekte zu gestatten. Die Aktivierung dieser Einstellung hat keine Auswirkungen auf vorhandene Zugriffspunkt- oder Bucket-Richtlinien.</p> <div data-bbox="430 1081 1507 1732" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> Important</p><p>Um diese Einstellung effektiv zu nutzen, empfehlen wir Ihnen, sie auf Konto-Ebene anzuwenden. Eine Bucket-Richtlinie kann Benutzern das Ändern der Block Public Access-Einstellungen eines Buckets gestatten. Daher könnten Benutzer mit der Berechtigung zum Ändern einer Bucket-Richtlinie eine Richtlinie einfügen, die es ihnen erlaubt, die Block Public Access-Einstellungen für den Bucket zu deaktivieren. Wenn diese Einstellung dann für das ganze Konto aktiviert wird statt für einen bestimmten Bucket, blockiert Amazon S3 öffentliche Richtlinien selbst dann, wenn ein Benutzer die Bucket-Richtlinie so ändert, dass diese Einstellung deaktiviert wird.</p></div>

Name	Beschreibung
RestrictPublicBuckets	<p>Wenn Sie diese Option auf einstellen, wird der Zugriff auf einen Access Point oder Bucket mit einer öffentlichen Richtlinie auf TRUE nur beschränkt AWS Dienstprinzipale und autorisierte Benutzer innerhalb des Kontos des Bucket-Besitzers und des Accounts des Access Point-Besitzers. Diese Einstellung blockiert jeglichen kontoübergreifenden Zugriff auf den Access Point oder Bucket (außer von AWS Service Principals), wobei Benutzer innerhalb des Kontos weiterhin den Access Point oder Bucket verwalten können.</p> <p>Die Aktivierung dieser Einstellung wirkt sich nicht auf vorhandene Zugriffspunkt- oder Bucket-Richtlinien aus – mit folgender Ausnahme: Amazon S3 blockiert den öffentlichen und kontenübergreifenden Zugriff, der von einer öffentlichen Zugriffspunkt- oder Bucket-Richtlinie abgeleitet wird, darunter auch die nicht-öffentliche Delegation auf bestimmte Konten.</p>

Important

- Aufrufe von GET Bucket ACL und GET Object ACL geben immer die effektiven Berechtigungen zurück, die für den angegebenen Bucket oder das angegebene Objekt vorhanden sind. Nehmen wir zum Beispiel an, ein Bucket hat einen BucketACL, der öffentlichen Zugriff gewährt, aber für den Bucket ist auch die IgnorePublicAcls Einstellung aktiviert. In diesem Fall gibt GET Bucket acl eine ACL zurück, die die Zugriffsberechtigungen widerspiegelt, die Amazon S3 erzwingt, und nicht die tatsächlichen, ACL die dem Bucket zugeordnet sind.
- Die Einstellungen zum Blockieren des öffentlichen Zugriffs ändern keine bestehenden Richtlinien oder ACLs. Daher führt das Entfernen einer Einstellung zum Blockieren des öffentlichen Zugriffs dazu, dass ein Bucket oder ein Objekt mit einer öffentlichen Richtlinie oder ACL wieder öffentlich zugänglich ist.

Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt

Um Operationen zum Blockieren des öffentlichen Zugriffs an einem Access Point durchzuführen, verwenden Sie den AWS CLI Dienst `s3control`.

Important

Beachten Sie, dass es derzeit nicht möglich ist, die Block Public Access-Einstellungen eines Zugriffspunkts nach dem Erstellen des Zugriffspunkts zu ändern. Die einzige Möglichkeit, Block Public Access-Einstellungen für einen Zugriffspunkt anzugeben, besteht darin, diese beim Erstellen des Zugriffspunkts einzuschließen.

Die Bedeutung von „öffentlich“

ACLs

Amazon S3 betrachtet einen Bucket oder ein Objekt als ACL öffentlich, wenn es Mitgliedern der vordefinierten `AuthenticatedUsers` Gruppen `AllUsers` oder Gruppen Berechtigungen gewährt. Weitere Informationen zu vordefinierten Gruppen finden Sie unter [Vordefinierte Gruppen in Amazon S3](#).

Bucket-Richtlinien

Bei der Evaluierung einer Bucket-Richtlinie beginnt Amazon S3 mit der Annahme, dass die Richtlinie öffentlich ist. Dann evaluiert es die Richtlinie, um festzustellen, ob sie als nicht-öffentlich eingestuft werden kann. Um als nicht öffentlich zu gelten, muss eine Bucket-Richtlinie nur Zugriff auf feste Werte gewähren (Werte, die keinen Platzhalter oder [AWS Identity and Access Management RichtlinienvARIABLE](#)) für eine oder mehrere der folgenden Optionen:

- Importieren in `&S3;` AWS Prinzipal, Benutzer, Rolle oder Dienstprinzipal (z. `B.aws:PrincipalOrgID`)
- Eine Reihe von Classless Inter-Domain Routings (CIDRs) unter Verwendung von `aws:SourceIp` Weitere Informationen zu CIDR finden Sie unter [RFC4632](#) auf der Editor-Website. RFC

Note

Bucket-Richtlinien, die abhängig vom `aws:SourceIp`-Bedingungsschlüssel Zugriff mit sehr breiten IP-Bereichen gewähren (z. B. `0.0.0.0/1`), werden als „öffentlich“ bewertet. Dies schließt Werte ein, die breiter sind als `/8` für IPv4 und `/32` für IPv6 (ohne RFC1918 private Bereiche). Die Funktion zum Blockieren des öffentlichen Zugriffs lehnt diese „öffentlichen“ Richtlinien ab und verhindert den kontoübergreifenden Zugriff auf Buckets, die diese „öffentlichen“ Richtlinien bereits verwenden.

- `aws:SourceArn`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `aws:userid`, außerhalb des Musters "AROLEID: *"
- `s3:DataAccessPointArn`

Note

Bei Verwendung in einer Bucket-Richtlinie kann dieser Wert einen Platzhalter für den Namen des Zugriffspunkts enthalten, ohne die Richtlinie öffentlich zu machen, solange die Konto-ID fixiert ist. Das Zulassen des Zugriffs auf `arn:aws:s3:us-west-2:123456789012:accesspoint/*` würde beispielsweise den Zugriff auf jeden Zugriffspunkt ermöglichen, der dem Konto 123456789012 in Region `us-west-2` zugeordnet ist, ohne die Bucket-Richtlinie öffentlich zu machen. Beachten Sie, dass sich dieses Verhalten von dem von Zugriffspunkt-Richtlinien unterscheidet. Weitere Informationen finden Sie unter [Zugriffspunkte](#).

- `s3:DataAccessPointAccount`

Weitere Informationen zu Bucket-Richtlinien finden Sie unter [Bucket-Richtlinien für Amazon S3](#).

Example : Öffentliche Bucket-Richtlinien

Nach diesen Regeln gelten die folgenden Beispielrichtlinien als öffentlich.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow"
}
```

```
{
  "Principal": "*",
  "Resource": "*",
```

```
"Action": "s3:PutObject",
"Effect": "Allow",
"Condition": { "StringLike": {"aws:SourceVpc": "vpc-*"}}
}
```

Diese Richtlinien können zu nicht-öffentlichen Richtlinien gemacht werden, indem einer der vorgenannten Bedingungsschlüssel unter Verwendung eines festen Wertes eingefügt wird. So können Sie beispielsweise die letzte oben angeführte Richtlinie zu einer nicht-öffentlichen Richtlinie machen, indem Sie `aws:SourceVpc` wie folgt auf einen festen Wert festlegen.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": {"StringEquals": {"aws:SourceVpc": "vpc-91237329"}}
}
```

So wertet Amazon S3 eine Bucket-Richtlinie aus, die Berechtigungen sowohl für den öffentlichen wie auch für den nicht-öffentlichen Zugriff enthält

Dieses Beispiel zeigt, wie Amazon S3 eine Bucket-Richtlinie bewertet, die Berechtigungen sowohl für den öffentlichen wie auch für den nicht-öffentlichen Zugriff enthält.

Nehmen wir an, dass ein Bucket eine Richtlinie aufweist, die den Zugriff auf einen Satz an festen Prinzipalen gewährt. Nach den zuvor beschriebenen Regeln gilt diese Richtlinie nicht als öffentlich. Wenn Sie daher die Einstellung `RestrictPublicBuckets` aktivieren, bleibt die Richtlinie wie geschrieben in Kraft, da `RestrictPublicBuckets` nur für Buckets mit öffentlichen Richtlinien gilt. Wenn Sie jedoch der Richtlinie eine öffentliche Anweisung hinzufügen, wird `RestrictPublicBuckets` für den Bucket wirksam. Es erlaubt nur AWS Service Principals und autorisierte Benutzer des Kontos des Bucket-Besitzers können auf den Bucket zugreifen.

Nehmen wir beispielsweise an, dass ein Bucket, der „Konto-1“ gehört, eine Richtlinie aufweist, die Folgendes enthält:

1. Eine Anweisung, die Zugriff gewährt auf AWS CloudTrail (was ist ein AWS Dienstleiter)
2. eine Anweisung, die Konto "Konto-2" Zugriff gewährt
3. eine Anweisung, die der Öffentlichkeit Zugriff gewährt, z. B. durch das Festlegen von `"Principal": "*" ohne einschränkende Condition`

Diese Richtlinie gilt wegen der dritten Anweisung als öffentlich. Wenn diese Richtlinie eingerichtet und `RestrictPublicBuckets` aktiviert ist, ermöglicht Amazon S3 den Zugriff nur von CloudTrail. Beachten Sie: Obwohl die zweite Anweisung nicht öffentlich ist, deaktiviert Amazon S3 den Zugriff durch „Account-2“. Das liegt daran, dass die dritte Anweisung die gesamte Richtlinie zu einer öffentlichen Richtlinie macht, so dass `RestrictPublicBuckets` gilt. Somit deaktiviert Amazon S3 den kontoübergreifenden Zugriff, obwohl die Richtlinie den Zugriff an ein bestimmtes Konto („Account-2“) delegiert. Wenn Sie aber die dritte Anweisung aus der Richtlinie entfernen, gilt die Richtlinie nicht als öffentlich, und `RestrictPublicBuckets` ist nicht mehr gültig. Somit erhält „Konto-2“ wieder Zugriff auf den Bucket, selbst wenn Sie `RestrictPublicBuckets` aktiviert lassen.

Zugriffspunkte

Amazon S3 wertet Block Public Access-Einstellungen für Zugriffspunkte geringfügig anders als für Buckets aus. Die Regeln, die Amazon S3 anwendet, um zu bestimmen, wann eine Zugriffspunkt-Richtlinie öffentlich ist, sind für Zugriffspunkte im Allgemeinen dieselben wie für Buckets, außer in den folgenden Situationen:

- Ein Access Point, der aus dem VPC Netzwerk stammt, gilt immer als nicht öffentlich, unabhängig vom Inhalt seiner Zugriffspunkt-Richtlinie.
- Eine Zugriffspunktrichtlinie, die Zugriff auf eine Gruppe von Zugriffspunkten unter Verwendung von `s3:DataAccessPointArn` gewährt, gilt als öffentlich. Beachten Sie, dass sich dieses Verhalten von dem von Bucket-Richtlinien unterscheidet. Beispielsweise wird eine Bucket-Richtlinie, die Zugriff auf `s3:DataAccessPointArn`-Werte gewährt, die `arn:aws:s3:us-west-2:123456789012:accesspoint/*` entsprechen, nicht als öffentlich betrachtet. Dieselbe Anweisung in einer Zugriffspunkt-Richtlinie würde jedoch den Zugriffspunkt öffentlich machen.

Verwenden von IAM Access Analyzer für S3 zur Überprüfung öffentlicher Buckets

Sie können IAM Access Analyzer for S3 verwenden, um Buckets mit BucketsACLs, Bucket-Richtlinien oder Access Point-Richtlinien zu überprüfen, die öffentlichen Zugriff gewähren. IAM Access Analyzer for S3 warnt Sie vor Buckets, die so konfiguriert sind, dass sie allen Personen im Internet oder auf anderen Websites Zugriff gewähren AWS-Konten, einschließlich AWS-Konten außerhalb Ihrer Organisation. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Ergebnisse, die die Quelle und die Ebene des öffentlichen oder freigegebenen Zugriffs melden.

In IAM Access Analyzer for S3 können Sie den gesamten öffentlichen Zugriff auf einen Bucket mit einem einzigen Klick blockieren. Sie können auch einen Drilldown in die Berechtigungseinstellungen

auf Bucket-Ebene ausführen, um detaillierte Zugriffsebenen zu konfigurieren. Für bestimmte und geprüfte Anwendungsfälle, die öffentlichen oder freigegebenen Zugriff erfordern, können Sie Ihre Absicht bestätigen und aufzeichnen, dass der Bucket öffentlich oder freigegeben bleibt, indem Sie die Ergebnisse für den Bucket archivieren.

In seltenen Fällen meldet IAM Access Analyzer for S3 möglicherweise keine Ergebnisse für einen Bucket, den eine Amazon S3 S3-Block-Bewertung für öffentlichen Zugriff als öffentlich meldet. Dies ist der Fall, da Amazon S3 Block Public Access Richtlinien für aktuelle Aktionen und potenzielle Aktionen, die in Zukunft hinzugefügt werden könnten und die dazu führen könnten, dass ein Bucket öffentlich wird, überprüft. Andererseits analysiert IAM Access Analyzer for S3 nur die aktuellen Aktionen, die für den Amazon S3 S3-Service bei der Bewertung des Zugriffsstatus angegeben wurden.

Weitere Informationen zu IAM Access Analyzer for S3 finden Sie unter [Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3](#).

Berechtigungen

Um die Funktionen von Amazon S3 Block Public Access zu nutzen, benötigen Sie die folgenden Berechtigungen:

Operation	Erforderliche Berechtigungen
GETStatus der Bucket-Richtlinie	s3:GetBucketPolicyStatus
GETEinstellungen für den Bucket „Öffentlichen Zugriff blockieren“	s3:GetBucketPublicAccessBlock
PUTBucket Einstellungen für den öffentlichen Zugriff blockieren	s3:PutBucketPublicAccessBlock
DELETEBucket Einstellungen für den öffentlichen Zugriff blockieren	s3:PutBucketPublicAccessBlock
GETEinstellungen für das Konto „Öffentlichen Zugriff blockieren“	s3:GetAccountPublicAccessBlock
PUTKonto Einstellungen für öffentlichen Zugriff sperren	s3:PutAccountPublicAccessBlock

Operation	Erforderliche Berechtigungen
DELETE Konto Einstellungen für öffentlichen Zugriff sperren	s3:PutAccountPublicAccessBlock
PUT Zugangspunkt Einstellungen für den öffentlichen Zugriff blockieren	s3:CreateAccessPoint

Note

Für die DELETE Operationen sind dieselben Berechtigungen erforderlich wie für die PUT Operationen. Für die DELETE Operationen gibt es keine separaten Berechtigungen.

Verwenden von Block Public Access

Weitere Informationen zur Konfiguration von Block Public Access für Ihr AWS-Konto, Ihre Amazon S3 S3-Buckets und Ihre Access Points finden Sie in den folgenden Themen:

- [Konfigurieren der Block-Public-Access-Einstellungen für Ihr Konto](#)
- [Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets](#)
- [Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt](#)

Konfigurieren der Block-Public-Access-Einstellungen für Ihr Konto

Die Amazon-S3-Block-Public-Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff.

Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Note

Einstellungen auf Kontoebene haben Vorrang vor Einstellungen für einzelne Objekte. Wenn Sie Ihr Konto so konfigurieren, dass der öffentliche Zugriff gesperrt wird, werden

alle Einstellungen für den öffentlichen Zugriff, die für einzelne Objekte in Ihrem Konto vorgenommen wurden, außer Kraft gesetzt.

Sie können die S3-Konsole verwenden, AWS CLI, AWS SDKs und REST API um die Einstellungen zum Sperren des öffentlichen Zugriffs für alle Buckets in Ihrem Konto zu konfigurieren. Weitere Informationen dazu finden Sie in den folgenden Abschnitten.

Informationen zum Konfigurieren von Block Public Access-Einstellungen für Ihre Buckets finden Sie unter [Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets](#). Weitere Hinweise zu Zugriffspunkten finden Sie unter [Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt](#).

Verwenden der S3-Konsole

Amazon S3 Block Public Access verhindert die Anwendung von Einstellungen, die den öffentlichen Zugriff auf Daten in S3-Buckets erlauben. In diesem Abschnitt wird beschrieben, wie Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs für alle S3-Buckets in Ihrem AWS-Konto. Weitere Informationen zum Sperren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

So bearbeiten Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs für alle S3-Buckets in einem AWS-Konto

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Block Public Access settings for this account (Block Public Access-Einstellung für dieses Konto).
3. Wählen Sie Bearbeiten, um die Einstellungen zum Sperren des öffentlichen Zugriffs für alle Buckets in Ihrem zu ändern AWS-Konto.
4. Wählen Sie die Einstellung aus, die Sie ändern möchten, und wählen Sie anschließend Save changes (Änderungen speichern) aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **confirm** ein. Wählen Sie anschließend Confirm (Bestätigen) aus, um Ihre Änderungen zu speichern.

Verwendung der AWS CLI

Sie können Amazon S3 Block Public Access verwenden über AWS CLI. Für weitere Informationen zur Einrichtung und Verwendung des AWS CLI, siehe [Was ist der AWS Command Line Interface?](#)

Account

Um Operationen zum Sperren des öffentlichen Zugriffs auf ein Konto durchzuführen, verwenden Sie den AWS CLI Diensts3control. Folgende Vorgänge auf Konto-Ebene verwenden diesen Service:

- PUT PublicAccessBlock (für ein Konto)
- GET PublicAccessBlock (für ein Konto)
- DELETE PublicAccessBlock (für ein Konto)

Weitere Informationen und Beispiele finden Sie [put-public-access-block](#) in der AWS CLI Referenz.

Verwendung der AWS SDKs

Java

Die folgenden Beispiele zeigen Ihnen, wie Sie Amazon S3 Block Public Access mit dem AWS SDK for Java um eine Blockkonfiguration für öffentlichen Zugriff auf ein Amazon S3 S3-Konto einzurichten.

```
AWSS3ControlClientBuilder controlClientBuilder =
    AWSS3ControlClientBuilder.standard();
controlClientBuilder.setRegion(<region>);
controlClientBuilder.setCredentials(<credentials>);

AWSS3Control client = controlClientBuilder.build();
client.putPublicAccessBlock(new PutPublicAccessBlockRequest()
    .withAccountId(<account-id>)
    .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
        .withIgnorePublicAcls(<value>)
        .withBlockPublicAcls(<value>)
        .withBlockPublicPolicy(<value>)
        .withRestrictPublicBuckets(<value>)));
```

⚠ Important

Dieses Beispiel trifft nur für Vorgänge auf Konto-Ebene zu, die die `AWSS3Control-Client`-Klasse verwenden. Sehen Sie sich für Vorgänge auf Bucket-Ebene das vorhergehende Beispiel an.

Other SDKs

Für Informationen zur Verwendung des anderen AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Mit dem REST API

Informationen zur Verwendung von Amazon S3 Block Public Access über die REST APIs finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference.

- Vorgänge auf Konto-Ebene
 - [PUT PublicAccessBlock](#)
 - [GET PublicAccessBlock](#)
 - [DELETE PublicAccessBlock](#)

Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets

Die Amazon-S3-Block-Public-Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff.

Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Sie können die S3-Konsole verwenden, AWS CLI, AWS SDKs, und REST API um öffentlichen Zugriff auf einen oder mehrere Buckets zu gewähren. Sie können den öffentlichen Zugriff auf Buckets, die bereits öffentlich sind, auch blockieren. Weitere Informationen dazu finden Sie in den folgenden Abschnitten.

Informationen zum Konfigurieren von Einstellungen zum Blockieren des öffentlichen Zugriffs für jeden Bucket in Ihrem Konto finden Sie unter [Konfigurieren der Block-Public-Access-Einstellungen für Ihr](#)

[Konto](#). Informationen zum Konfigurieren von Block Public Access für Zugriffspunkte finden Sie unter [Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt](#).

Verwenden der S3-Konsole

Amazon S3 Block Public Access verhindert die Anwendung von Einstellungen, die den öffentlichen Zugriff auf Daten in S3-Buckets erlauben. In diesem Abschnitt wird erläutert, wie Sie die Block-Public-Access-Einstellungen für einen oder mehrere S3-Buckets bearbeiten. Informationen zum Sperren des öffentlichen Zugangs finden Sie unter AWS CLI, AWS SDKs, und Amazon S3 REST APIs, siehe [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

In der Liste der Buckets können Sie sehen, ob Ihr Bucket öffentlich zugänglich ist. In der Spalte Zugriff kennzeichnet Amazon S3 die Berechtigungen für einen Bucket wie folgt:

- Public (Öffentlich) – Alle Benutzer haben Zugriff auf eines oder mehrere der folgenden Elemente: Objekte auflisten, Objekte schreiben, Lese- und Schreibberechtigungen.
- Objects can be public (Objekte können öffentlich sein) – Der Bucket ist nicht öffentlich, aber jeder, der über die entsprechenden Berechtigungen verfügt, kann den öffentlichen Zugriff auf Objekte gewähren.
- Buckets and objects not public (Buckets und Objekte sind nicht öffentlich) – Für den Bucket und die Objekte gilt kein öffentlicher Zugriff.
- Nur autorisierte Benutzer dieses Kontos — Der Zugriff ist auf IAM Benutzer und Rollen in diesem Konto beschränkt und AWS Service Principals, weil es eine Richtlinie gibt, die öffentlichen Zugriff gewährt.

Sie können Bucket-Suchen auch nach Zugriffstyp filtern. Wählen Sie aus der Dropdown-Liste neben der Leiste Search for buckets (Nach Buckets suchen) einen Zugriffstyp aus.

Wenn beim Auflisten Ihrer Buckets mit ihren Einstellungen für den öffentlichen Zugriff die Fehlermeldung `Error` angezeigt wird, verfügen Sie möglicherweise nicht über die erforderlichen Berechtigungen. Stellen Sie sicher, dass Sie Ihrer Benutzer- oder Rollenrichtlinie die folgenden Berechtigungen hinzugefügt haben:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
```

```
s3:ListAccessPoints  
s3:ListAllMyBuckets
```

In einigen seltenen Fällen können Anfragen auch aufgrund eines fehlschlagen AWS-Region Ausfall.

Bearbeiten der Amazon-S3-Block-Public-Access-Einstellungen für einen einzelnen S3-Bucket

Führen Sie die folgenden Schritte aus, wenn Sie die Einstellungen für den öffentlichen Zugriff für einen einzelnen S3-Bucket ändern müssen.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Bucket-Name den Namen des gewünschten Buckets aus.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie Edit (Bearbeiten) aus, um die Einstellungen für den öffentlichen Zugriff für den Bucket zu ändern. Weitere Informationen zu den vier Amazon-S3-Block-Public-Access-Einstellungen finden Sie unter [Block Public Access-Einstellungen](#).
5. Wählen Sie die Einstellung aus, die Sie ändern möchten, und wählen Sie anschließend Save (Speichern) aus.
6. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **confirm** ein. Wählen Sie anschließend Confirm (Bestätigen) aus, um Ihre Änderungen zu speichern.

Sie können beim Erstellen eines Buckets die Amazon S3 Block Public Access-Einstellungen ändern. Weitere Informationen finden Sie unter [Erstellen eines Buckets](#).

Verwendung der AWS CLI

Um den öffentlichen Zugriff auf einen Bucket zu blockieren oder den Block für den öffentlichen Zugriff zu löschen, verwenden Sie den AWS CLI Diensts3api. Folgende Vorgänge auf Bucket-Ebene verwenden diesen Service:

- PUT PublicAccessBlock (für einen Eimer)
- GET PublicAccessBlock (für einen Eimer)
- DELETE PublicAccessBlock (für einen Eimer)
- GET BucketPolicyStatus

Weitere Informationen und Beispiele finden Sie [put-public-access-block](#) in der AWS CLI Referenz.

Verwendung der AWS SDKs

Java

```
AmazonS3 client = AmazonS3ClientBuilder.standard()
    .withCredentials(<credentials>)
    .build();

client.setPublicAccessBlock(new SetPublicAccessBlockRequest()
    .withBucketName(<bucket-name>)
    .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
        .withBlockPublicAcls(<value>)
        .withIgnorePublicAcls(<value>)
        .withBlockPublicPolicy(<value>)
        .withRestrictPublicBuckets(<value>)));
```

Important

Dieses Beispiel trifft nur für Vorgänge auf Bucket-Ebene zu, die die AmazonS3-Client-Klasse verwenden. Sehen Sie sich für Vorgänge auf Konto-Ebene das folgende Beispiel an.

Other SDKs

Für Informationen zur Verwendung des anderen AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Mit dem REST API

Informationen zur Verwendung von Amazon S3 Block Public Access über die REST APIs finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference.

- Vorgänge auf Bucket-Ebene
 - [PUT PublicAccessBlock](#)
 - [GET PublicAccessBlock](#)
 - [DELETE PublicAccessBlock](#)

- [GET BucketPolicyStatus](#)

Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3

IAM Access Analyzer for S3 warnt Sie vor S3-Buckets, die so konfiguriert sind, dass sie allen Personen im Internet oder auf anderen Websites Zugriff gewähren AWS-Konten, einschließlich AWS-Konten außerhalb Ihrer Organisation. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Ergebnisse bezüglich der Quelle und der Ebene des öffentlichen oder freigegebenen Zugriffs. IAM Access Analyzer for S3 könnte beispielsweise anzeigen, dass für einen Bucket Lese- oder Schreibzugriff über eine Bucket-Zugriffskontrollliste (ACL), eine Bucket-Richtlinie, eine Multi-Region-Access-Point-Richtlinie oder eine Access-Point-Richtlinie gewährt wird. Mit diesen Erkenntnissen können Sie sofortige und präzise Korrekturmaßnahmen ergreifen, um den Bucket-Zugriff wie beabsichtigt wiederherzustellen.

Bei der Überprüfung eines gefährdeten Buckets in IAM Access Analyzer for S3 können Sie den gesamten öffentlichen Zugriff auf den Bucket mit einem einzigen Klick blockieren. Wir empfehlen Ihnen, den gesamten Zugriff auf Ihre Buckets zu blockieren, es sei denn, Sie benötigen öffentlichen Zugriff, um einen bestimmten Anwendungsfall zu unterstützen. Bevor Sie den gesamten öffentlichen Zugriff blockieren, stellen Sie sicher, dass Ihre Anwendungen ohne öffentlichen Zugriff weiterhin ordnungsgemäß funktionieren. Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

Sie können auch einen Drilldown in die Berechtigungseinstellungen auf Bucket-Ebene ausführen, um detaillierte Zugriffsebenen zu konfigurieren. Für bestimmte und überprüfte Anwendungsfälle, die öffentlichen Zugriff erfordern, wie statisches Website-Hosting, öffentliche Downloads oder kontenübergreifende Freigabe, können Sie Ihre Absicht bestätigen und aufzeichnen, dass der Bucket öffentlich oder freigegeben bleibt, indem Sie die Ergebnisse für den Bucket archivieren. Diese Bucket-Konfigurationen sind jederzeit aufrufbar und änderbar. Sie können Ihre Ergebnisse zu Prüfungszwecken auch als CSV Bericht herunterladen.

IAM Access Analyzer for S3 ist ohne zusätzliche Kosten auf der Amazon S3 S3-Konsole verfügbar. IAM Access Analyzer for S3 wird unterstützt von AWS Identity and Access Management (IAM) IAM Access Analyzer. Um IAM Access Analyzer for S3 in der Amazon S3 S3-Konsole zu verwenden, müssen Sie die IAM Konsole aufrufen und IAM Access Analyzer für jede Region aktivieren.

Weitere Informationen zu IAM Access Analyzer finden Sie unter [Was ist IAM Access Analyzer?](#) im IAM Benutzerhandbuch. Weitere Informationen zu IAM Access Analyzer for S3 finden Sie in den folgenden Abschnitten.

Important

- IAMAccess Analyzer für S3 erfordert einen Analyzer auf Kontoebene. Um IAM Access Analyzer für S3 verwenden zu können, müssen Sie IAM Access Analyzer aufrufen und einen Analyzer erstellen, der über ein Konto als Vertrauenszone verfügt. Weitere Informationen finden Sie unter [Enabling IAM Access Analyzer](#) im IAMBenutzerhandbuch.
- IAMAccess Analyzer for S3 analysiert nicht die Access Point-Richtlinie, die kontoübergreifenden Access Points zugeordnet ist. Dieses Verhalten tritt auf, weil sich der Zugriffspunkt und seine Richtlinie außerhalb der Vertrauenszone, d. h. des Kontos, befinden. Buckets, die den Zugriff an einen kontoübergreifenden Zugriffspunkt delegieren, werden unter Buckets mit öffentlichem Zugriff aufgeführt, sofern Sie die Einstellung `RestrictPublicBuckets` zum Blockieren des öffentlichen Zugriffs nicht auf den Bucket oder das Konto angewendet haben. Wenn Sie die Einstellung „Öffentlichen Zugriff `RestrictPublicBuckets` blockieren“ anwenden, wird der Bucket unter Buckets mit Zugriff von anderen Benutzern angezeigt AWS-Konten — einschließlich Drittanbieter AWS-Konten.
- Wenn eine Bucket-Richtlinie oder ein Bucket hinzugefügt oder geändert ACL wird, generiert und aktualisiert IAM Access Analyzer die Ergebnisse auf der Grundlage der Änderung innerhalb von 30 Minuten. Ergebnisse im Zusammenhang mit den Einstellungen zum Blockieren des öffentlichen Zugriffs auf Kontoebene werden möglicherweise erst bis zu 6 Stunden, nachdem Sie die Einstellungen geändert haben, generiert oder aktualisiert. Ergebnisse, die sich auf Multi-Regions-Zugriffspunkte beziehen, können bis zu sechs Stunden nach dem Erstellen, Löschen oder Ändern der Richtlinie des Multi-Region Access Points nicht generiert oder aktualisiert werden.

Topics

- [Welche Informationen bietet IAM Access Analyzer for S3?](#)
- [IAMAccess Analyzer für S3 aktivieren](#)
- [Blockieren des gesamten öffentlichen Zugriffs](#)
- [Überprüfen und Ändern des Bucket-Zugriffs](#)
- [Archivieren von Bucket-Ergebnissen](#)
- [Aktivieren eines archivierten Bucket-Ergebnisses](#)
- [Anzeigen von Ergebnisdetails](#)

- [Ein IAM Access Analyzer for S3-Bericht wird heruntergeladen](#)

Welche Informationen bietet IAM Access Analyzer for S3?

IAM Access Analyzer for S3 liefert Ergebnisse für Buckets, auf die außerhalb Ihres AWS-Konto. Buckets, die unter Buckets with public access aufgeführt sind, können von jedem im Internet abgerufen werden. Wenn IAM Access Analyzer for S3 öffentliche Buckets identifiziert, wird oben auf der Seite auch eine Warnung angezeigt, die Ihnen die Anzahl der öffentlichen Buckets in Ihrer Region anzeigt. Buckets, die unter Buckets mit Zugriff von anderen Anbietern aufgeführt sind AWS-Konten — auch von Drittanbietern AWS-Konten werden unter bestimmten Bedingungen mit anderen geteilt AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation.

IAM Access Analyzer for S3 stellt für jeden Bucket die folgenden Informationen bereit:

- Bucket-Name
- Von Access Analyzer entdeckt - Als IAM Access Analyzer for S3 den öffentlichen oder gemeinsam genutzten Bucket-Zugriff entdeckte.
- Gemeinsam genutzt — wie der Bucket gemeinsam genutzt wird — über eine Bucket-Richtlinie, einen BucketACL, eine Multi-Region-Access-Point-Richtlinie oder eine Access-Point-Richtlinie. Multi-Region Access Points und kontoübergreifende Zugriffspunkte werden unter Zugriffspunkten aufgeführt. Ein Bucket kann sowohl über Richtlinien als auch gemeinsam genutzt werden. ACLs Wenn Sie die Quelle für den Bucket-Zugriff suchen und überprüfen möchten, können die Informationen in dieser Spalte als Ausgangspunkt für sofortige und präzise Korrekturmaßnahmen dienen.
- Status - Der Status der Erkenntnisse zu dem Bucket. IAM Access Analyzer for S3 zeigt Ergebnisse für alle öffentlichen und gemeinsam genutzten Buckets an.
 - Active (Aktiv) – Ergebnis wurde nicht überprüft.
 - Archived (Archiviert) – Das Ergebnis wurde wie vorgesehen überprüft und bestätigt.
 - Alle - Alle Ergebnisse für Buckets, die öffentlich sind oder mit anderen geteilt werden AWS-Konten, einschließlich AWS-Konten außerhalb Ihrer Organisation.
- Access level (Zugriffsebene) – Gewährte Zugriffsberechtigungen für den Bucket:
 - List (Liste) – Auflistung der Ressourcen.
 - Read (Lesen) – Lesen aber kein Bearbeiten von Ressourceninhalten und -attributen.
 - Write (Schreiben) – Erstellen, Löschen oder Ändern von Ressourcen.
 - Permissions (Berechtigungen) – Erteilen oder ändern der Ressourcenberechtigungen.

- Markierungen – Aktualisieren der einer Ressource zugeordneten Markierungen.

IAM Access Analyzer für S3 aktivieren

Um IAM Access Analyzer für S3 verwenden zu können, müssen Sie die folgenden Voraussetzungen erfüllen.

1. Sie müssen die erforderlichen Berechtigungen erteilen.

Weitere Informationen finden Sie im IAM Benutzerhandbuch unter Für [die Verwendung von IAM Access Analyzer erforderliche Berechtigungen](#).

2. Besuchen Sie IAM, um für jede Region, in der Sie IAM Access Analyzer verwenden möchten, einen Analyzer auf Kontoebene zu erstellen.

IAM Access Analyzer für S3 erfordert einen Analyzer auf Kontoebene. Um IAM Access Analyzer für S3 verwenden zu können, müssen Sie einen Analyzer erstellen, der über ein Konto als Vertrauenszone verfügt. Weitere Informationen finden Sie unter [Enabling IAM Access Analyzer](#) im IAM Benutzerhandbuch.

Blockieren des gesamten öffentlichen Zugriffs

Wenn Sie den gesamten Zugriff auf einen Bucket mit einem einzigen Klick blockieren möchten, können Sie die Schaltfläche **Allen öffentlichen Zugriff blockieren** in IAM Access Analyzer für S3 verwenden. Wenn Sie den gesamten öffentlichen Zugriff auf einen Bucket blockieren, wird kein öffentlicher Zugriff gewährt. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren, es sei denn, Sie benötigen öffentlichen Zugriff, um einen bestimmten und verifizierten Anwendungsfall zu unterstützen. Bevor Sie den gesamten öffentlichen Zugriff blockieren, stellen Sie sicher, dass Ihre Anwendungen ohne öffentlichen Zugriff weiterhin ordnungsgemäß funktionieren.

Wenn Sie nicht den gesamten öffentlichen Zugriff auf Ihren Bucket blockieren möchten, können Sie Ihre **Block Public Access**-Einstellungen in der Amazon-S3-Konsole bearbeiten, um detaillierte Zugriffsebenen für Ihre Buckets zu konfigurieren. Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

In seltenen Fällen meldet IAM Access Analyzer für S3 möglicherweise keine Ergebnisse für einen Bucket, den eine Amazon S3 S3-Block-Bewertung für öffentlichen Zugriff als öffentlich meldet. Dies ist der Fall, da Amazon S3 Block Public Access Richtlinien für aktuelle Aktionen und potenzielle Aktionen, die in Zukunft hinzugefügt werden könnten und die dazu führen könnten, dass ein Bucket

öffentlich wird, überprüft. Andererseits analysiert IAM Access Analyzer for S3 nur die aktuellen Aktionen, die für den Amazon S3 S3-Service bei der Bewertung des Zugriffsstatus angegeben wurden.

Um den gesamten öffentlichen Zugriff auf einen Bucket mithilfe von IAM Access Analyzer for S3 zu blockieren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite unter Dashboards die Option Access Analyzer for S3 aus.
3. Wählen Sie in Access Analyzer for S3 einen Bucket aus.
4. Wählen Sie Block all public access (Öffentlichen Zugriff blockieren) aus.
5. Um zu bestätigen, dass Sie den gesamten öffentlichen Zugriff auf den Bucket blockieren möchten, geben Sie unter Block all public access (bucket settings) (Öffentlichen Zugriff blockieren) (Bucket-Einstellungen) **confirm** ein.

Amazon S3 blockiert den gesamten öffentlichen Zugriff auf Ihren Bucket. Der Status des Buckets, bei dem Aktualisierungen gefunden wurden, lautet „Gelöst“, und der Bucket verschwindet aus der Liste von IAM Access Analyzer für S3. Wenn Sie die gelösten Buckets überprüfen möchten, öffnen Sie IAM Access Analyzer in der [IAMKonsole](#).

Überprüfen und Ändern des Bucket-Zugriffs

Wenn Sie nicht beabsichtigen, der Öffentlichkeit oder anderen Personen Zugriff zu gewähren AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation, können Sie den Bucket, die Bucket-RichtlinieACL, die Multi-Region-Zugriffspunkt-Richtlinie oder die Zugriffspunktrichtlinie ändern, um den Zugriff auf den Bucket zu entfernen. In der Spalte „Geteilt durch“ werden alle Quellen für den Bucket-Zugriff angezeigt: Bucket-RichtlinieACL, Bucket- und/oder Access-Point-Richtlinie. Multi-Region Access Points und kontoübergreifende Zugriffspunkte werden unter Zugriffspunkten aufgeführt.

Um eine Bucket-Richtlinie, einen Bucket, einen Access Point für mehrere Regionen oder eine Zugriffspunkt-Richtlinie zu überprüfen und zu ändern ACL

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.

3. Um zu sehen, ob öffentlicher Zugriff oder gemeinsamer Zugriff über eine Bucket-Richtlinie, einen BucketACL, eine Multi-Region-Access-Point-Richtlinie oder eine Access-Point-Richtlinie gewährt wird, schauen Sie in der Spalte Shared through nach.
4. Wählen Sie unter Buckets den Namen des Buckets mit der Bucket-Richtlinie, dem BucketACL, der Multi-Region-Zugriffspunkt-Richtlinie oder der Zugriffspunktrichtlinie aus, die Sie ändern oder überprüfen möchten.
5. Wenn Sie einen Bucket ändern oder anzeigen möchten: ACL
 - a. Wählen Sie Permissions (Berechtigungen).
 - b. Wählen Sie Access Control List.
 - c. Überprüfen Sie Ihren Bucket ACL und nehmen Sie bei Bedarf Änderungen vor.

Weitere Informationen finden Sie unter [Konfiguration ACLs](#).

6. Wenn Sie eine Bucket-Richtlinie ändern oder überprüfen möchten, gehen Sie folgendermaßen vor:
 - a. Wählen Sie Permissions (Berechtigungen).
 - b. Wählen Sie Bucket Policy aus.
 - c. Überprüfen oder ändern Sie Ihre Bucket-Richtlinie nach Bedarf.

Weitere Informationen finden Sie unter [Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole](#).

7. Wenn Sie eine Multi-Regions-Zugriffspunkt-Richtlinie ändern oder anzeigen, gehen Sie folgendermaßen vor:
 - a. Wählen Sie Multiregionaler Zugriffspunkt.
 - b. Wählen Sie den multiregionalen Zugriffspunktsnamen.
 - c. Überprüfen oder ändern Sie die Multi-Regions-Zugriffspunkt-Richtlinie nach Bedarf.

Weitere Informationen finden Sie unter [Berechtigungen](#).

8. Wenn Sie eine Zugriffspunkt-Richtlinie überprüfen oder ändern möchten, gehen Sie folgendermaßen vor:
 - a. Wählen Sie Access Points (Zugriffspunkte).
 - b. Wählen Sie den Namen des Zugriffspunkts aus.
 - c. Überprüfen oder ändern Sie den Zugriff nach Bedarf.

Weitere Informationen finden Sie unter [Nutzen von Amazon S3-Zugriffspunkten mit der Amazon S3-Konsole](#).

Wenn Sie einen BucketACL, eine Bucket-Richtlinie oder eine Zugriffspunkt-Richtlinie bearbeiten oder entfernen, um öffentlichen oder gemeinsamen Zugriff zu entfernen, wird der Status der Bucket-Ergebnisse auf „Gelöst“ aktualisiert. Die aufgelösten Bucket-Ergebnisse werden nicht mehr in der Liste von IAM Access Analyzer for S3 angezeigt, Sie können sie jedoch in IAM Access Analyzer anzeigen.

Archivieren von Bucket-Ergebnissen

Wenn ein Bucket der Öffentlichkeit oder anderen Personen Zugriff gewährt AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation, können Sie die Ergebnisse für den Bucket archivieren, um einen bestimmten Anwendungsfall zu unterstützen (z. B. eine statische Website, öffentliche Downloads oder kontoübergreifendes Teilen). Wenn Sie Bucket-Ergebnisse archivieren, bestätigen und verzeichnen Sie Ihre Absicht, dass der Bucket öffentlich oder freigegeben bleiben soll. Archivierte Bucket-Ergebnisse verbleiben in Ihrer IAM Access Analyzer for S3-Liste, sodass Sie immer wissen, welche Buckets öffentlich oder gemeinsam genutzt werden.

Um Bucket-Ergebnisse in IAM Access Analyzer für S3 zu archivieren

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.
3. Wählen Sie in IAM Access Analyzer for S3 einen aktiven Bucket aus.
4. Um Ihre Absicht zu bestätigen, dass auf diesen Bucket von der Öffentlichkeit oder anderen zugegriffen werden soll AWS-Konten, auch für Konten außerhalb Ihrer Organisation, wählen Sie Archivieren.
5. Geben Sie **confirm** ein und wählen Sie Archiv (Archivieren).

Aktivieren eines archivierten Bucket-Ergebnisses

Nachdem Sie Ergebnisse archiviert haben, können Sie sie jederzeit erneut einsehen und ihren Status wieder auf aktiv ändern, wodurch angegeben wird, dass für den Bucket eine weitere Überprüfung erforderlich ist.

Um die Suche nach archivierten Buckets in IAM Access Analyzer for S3 zu aktivieren

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.
3. Wählen Sie die archivierten Bucket-Ergebnisse aus.
4. Wählen Sie Mark as active (Als aktiv markieren) aus.

Anzeigen von Ergebnisdetails

Wenn Sie weitere Informationen zu einem Bucket benötigen, können Sie den Bucket öffnen und Details in IAM Access Analyzer auf der [IAMKonsole](#) suchen.

Um Details zu den Ergebnissen in IAM Access Analyzer für S3 anzuzeigen

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.
3. Wählen IAM Sie in Access Analyzer for S3 einen Bucket aus.
4. Wählen Sie die Option View details aus.

Die Ergebnisdetails werden in IAM Access Analyzer auf der [IAMKonsole](#) geöffnet.

Ein IAM Access Analyzer for S3-Bericht wird heruntergeladen

Sie können Ihre Bucket-Ergebnisse als CSV Bericht herunterladen, den Sie für Prüfungszwecke verwenden können. Der Bericht enthält dieselben Informationen, die Sie in IAM Access Analyzer for S3 auf der Amazon S3 S3-Konsole sehen.

So laden Sie einen Bericht herunter

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite Access Analyzer for S3 aus.
3. Wählen Sie im Filter „Region“ die Region aus.

IAMAccess Analyzer for S3-Updates zeigt Buckets für die gewählte Region an.

4. Wählen Sie Download Report (Bericht herunterladen) aus.

Ein CSV Bericht wird generiert und auf Ihrem Computer gespeichert.

Überprüfen der Bucket-Eigentümerschaft mit Bucket-Eigentümer-Bedingung

Die Bedingung des Amazon S3 S3-Bucket-Besitzers stellt sicher, dass die Buckets, die Sie in Ihren S3-Vorgängen verwenden, zu den AWS-Konten das erwarten Sie.

Die meisten S3-Vorgänge lesen oder schreiben in bestimmte S3-Buckets. Zu diesen Vorgängen gehören das Hochladen, Kopieren und Herunterladen von Objekten, das Abrufen oder Ändern von Bucket-Konfigurationen und das Abrufen oder Ändern von Objektkonfigurationen. Wenn Sie diese Vorgänge ausführen, geben Sie den Bucket an, den Sie verwenden möchten, indem Sie seinen Namen in die Anforderung einbeziehen. Um beispielsweise ein Objekt aus S3 abzurufen, stellen Sie eine Anforderung, die den Namen eines Buckets und den Objektschlüssel angibt, der aus diesem Bucket abgerufen werden soll.

Da Amazon S3 Buckets anhand ihrer Namen identifiziert, könnte eine Anwendung, die einen falschen Bucket-Namen in einer Anforderung verwendet, versehentlich Vorgänge gegen einen anderen Bucket als erwartet ausführen. Um unbeabsichtigte Bucket-Interaktionen in solchen Situationen zu vermeiden, können Sie die Bucket-Eigentümer-Bedingung verwenden. Mit der Bedingung „Bucket-Besitzer“ können Sie überprüfen, ob der Ziel-Bucket dem erwarteten Benutzer gehört AWS-Konto und bietet so eine zusätzliche Gewissheit, dass Ihre S3-Operationen die von Ihnen beabsichtigten Auswirkungen haben.

Themen

- [Wann die Bucket-Eigentümer-Bedingung verwendet werden sollte](#)
- [Verifizieren eines Bucket-Eigentümers](#)
- [Beispiele](#)
- [Beschränkungen und Einschränkungen](#)

Wann die Bucket-Eigentümer-Bedingung verwendet werden sollte

Wir empfehlen, die Bucket-Eigentümer-Bedingung zu verwenden, wenn Sie einen unterstützten S3-Vorgang ausführen und die Konto-ID des erwarteten Bucket-Eigentümers kennen. Die Bucket-Eigentümer-Bedingung ist für alle S3-Objekt-Vorgänge und die meisten S3-Bucketvorgänge verfügbar. Eine Liste der S3-Vorgänge, die die Bucket-Eigentümerbedingung nicht unterstützen, finden Sie unter [Beschränkungen und Einschränkungen](#).

Um die Vorteile der Verwendung der Bucket-Eigentümer-Bedingung zu erkennen, stellen Sie sich das folgende Szenario vor AWS Kunde Bea:

1. Bea entwickelt eine Anwendung, die Amazon S3 verwendet. Während der Entwicklung verwendet Bea sie nur zum Testen AWS-Konto um einen Bucket mit dem Namen zu erstellen und ihre Anwendung so zu konfigurieren `bea-data-test`, dass sie Anfragen an sie stellt. `bea-data-test`
2. Bea stellt ihre Anwendung bereit, vergisst jedoch, die Anwendung neu zu konfigurieren, sodass sie in ihrer Produktion einen Bucket verwendet AWS-Konto.
3. In der Produktion stellt Beas Anwendung Anforderungen an `bea-data-test`, die erfolgreich sind. Dies führt dazu, dass Produktionsdaten in den Bucket im Testkonto von Bea geschrieben werden.

Bea kann sich vor solchen Situationen schützen, indem sie die Bucket-Eigentümer-Bedingung verwendet. Mit der Bedingung „Besitzer des Buckets“ kann Bea Folgendes einbeziehen AWS-Konto ID des voraussichtlichen Bucket-Besitzers in ihren Anfragen. Amazon S3 überprüft dann die Konto-ID des Bucket-Eigentümers, bevor es eine Anforderung bearbeitet. Wenn der tatsächliche Bucket-Eigentümer nicht mit dem erwarteten Bucket-Eigentümer übereinstimmt, schlägt die Anforderung fehl.

Wenn Bea die Bucket-Eigentümer-Bedingung verwendet, führt das zuvor beschriebene Szenario nicht dazu, dass die Anwendung von Bea versehentlich in einen Test-Bucket schreibt. Stattdessen schlagen die Anforderungen, die ihre Anwendung in Schritt 3 stellt, mit einer Fehlermeldung `Access Denied` fehl. Durch die Verwendung der Bedingung „Bucket Owner“ trägt Bea dazu bei, das Risiko einer versehentlichen Interaktion mit Buckets im falschen Format zu vermeiden AWS-Konto.

Verifizieren eines Bucket-Eigentümers

Um die Bucket-Eigentümer-Bedingung zu verwenden, schließen Sie in ihre Anforderung einen Parameter ein, der den erwarteten Bucket-Eigentümer angibt. Die meisten S3-Vorgänge betreffen nur einen einzelnen Bucket und erfordern nur diesen einzelnen Parameter, um die Bucket-Eigentümer-Bedingung zu verwenden. Für `CopyObject`-Vorgänge gibt dieser erste Parameter den erwarteten Eigentümer des Ziel-Buckets an, und Sie schließen einen zweiten Parameter ein, um den erwarteten Eigentümer des Quell-Buckets anzugeben.

Wenn Sie eine Anforderung stellen, die einen Bucket-Eigentümer-Bedingungsparameter enthält, überprüft S3 vor der Verarbeitung der Anforderung die Konto-ID des Bucket-Eigentümers mit dem angegebenen Parameter. Wenn der Parameter mit der Konto-ID des Bucket-Eigentümers übereinstimmt, verarbeitet S3 die Anforderung. Wenn der Parameter nicht mit der Konto-ID des Bucket-Eigentümers übereinstimmt, schlägt die Anforderung mit einer Fehlermeldung `Access Denied` fehl.

Sie können den Bucket Owner Condition zusammen mit dem AWS Command Line Interface (AWS CLI), AWS SDKs und Amazon S3 REST APIs. Wenn Sie die Bucket-Eigentümer-Bedingung mit dem AWS CLI und Amazon S3 REST APIs, verwenden Sie die folgenden Parameternamen.

Zugriffsmethode	Parameter für Nicht-Kopiervorgänge	Quellparameter für Kopiervorgang	Zielparameter für Kopiervorgang
AWS CLI	<code>--expected-bucket-owner</code>	<code>--expected-source-bucket-owner</code>	<code>--expected-bucket-owner</code>
Amazon S3 REST APIs	Header <code>x-amz-expected-bucket-owner</code>	Header <code>x-amz-source-expected-bucket-owner</code>	Header <code>x-amz-expected-bucket-owner</code>

Die Parameternamen, die für die Verwendung der Bucket-Eigentümer-Bedingung mit dem AWS SDKs variieren je nach Sprache. Informationen zur Bestimmung der erforderlichen Parameter finden Sie in der SDK Dokumentation für Ihre gewünschte Sprache. Sie finden die SDK Dokumentation unter [Tools to Build on AWS](#).

Beispiele

Die folgenden Beispiele zeigen, wie Sie die Bucket-Besitzer-Bedingung in Amazon S3 implementieren können, indem Sie AWS CLI oder das AWS SDK for Java 2.x.

Example

Beispiel: Ein Objekt hochladen

Im folgenden Beispiel wird ein Objekt in den S3-Bucket hochgeladen `amzn-s3-demo-bucket1`, wobei die Bucket-Owner-Bedingung verwendet `amzn-s3-demo-bucket1` wird, um sicherzustellen, dass es Eigentum von AWS-Konto 111122223333.

AWS CLI

```
aws s3api put-object \
    --bucket amzn-s3-demo-bucket1 --key exampleobject --
body example_file.txt \
    --expected-bucket-owner 111122223333
```

AWS SDK for Java 2.x

```
public void putObjectExample() {
    S3Client s3Client = S3Client.create();
    PutObjectRequest request = PutObjectRequest.builder()
        .bucket("amzn-s3-demo-bucket1")
        .key("exampleobject")
        .expectedBucketOwner("111122223333")
        .build();
    Path path = Paths.get("example_file.txt");
    s3Client.putObject(request, path);
}
```

Example

Beispiel: Kopieren eines Objekts

Im folgenden Beispiel wird das Objekt `object1` vom S3-Bucket `amzn-s3-demo-bucket1` in den S3-Bucket `amzn-s3-demo-bucket2` kopiert. Dabei wird die Bucket-Eigentümer-Bedingung verwendet, um sicherzustellen, dass die Buckets den erwarteten Konten gemäß der folgenden Tabelle gehören.

Bucket	Erwarteter Eigentümer
<code>amzn-s3-demo-bucket1</code>	111122223333
<code>amzn-s3-demo-bucket2</code>	444455556666

AWS CLI

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket1/object1 \
    --bucket amzn-s3-demo-bucket2 --key object1copy \
    --expected-source-bucket-owner 111122223333 --expected-
bucket-owner 444455556666
```

AWS SDK for Java 2.x

```
public void copyObjectExample() {
```

```
S3Client s3Client = S3Client.create();
CopyObjectRequest request = CopyObjectRequest.builder()
    .copySource("amzn-s3-demo-bucket1/object1")
    .destinationBucket("amzn-s3-demo-bucket2")
    .destinationKey("object1copy")
    .expectedSourceBucketOwner("111122223333")
    .expectedBucketOwner("444455556666")
    .build();
s3Client.copyObject(request);
}
```

Example

Beispiel: Abrufen einer Bucket-Richtlinie

Im folgenden Beispiel wird die Zugriffsrichtlinie für den S3-Bucket abgerufen *amzn-s3-demo-bucket1*, wobei die Bedingung bucket owner verwendet wird, um sicherzustellen, dass dieser Bucket *amzn-s3-demo-bucket1* gehört AWS-Konto 111122223333.

AWS CLI

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket1 --expected-bucket-owner 111122223333
```

AWS SDK for Java 2.x

```
public void getBucketPolicyExample() {
    S3Client s3Client = S3Client.create();
    GetBucketPolicyRequest request = GetBucketPolicyRequest.builder()
        .bucket("amzn-s3-demo-bucket1")
        .expectedBucketOwner("111122223333")
        .build();
    try {
        GetBucketPolicyResponse response = s3Client.getBucketPolicy(request);
    }
    catch (S3Exception e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
}
```

Beschränkungen und Einschränkungen

Die Amazon-S3-Bucket-Eigentümer-Bedingung hat die folgenden Einschränkungen und Grenzen:

- Der Wert des Bedingungsparameters Bucket Owner muss ein sein AWS-Konto ID (12-stelliger numerischer Wert). Service-Prinzipale werden nicht unterstützt.
- Die Bedingung „Bucket-Besitzer“ ist weder für [CreateBucketListBuckets](#), noch für andere der in enthaltenen Operationen verfügbar [AWS S3-Steuerung](#). Amazon S3 ignoriert alle Bucket-Eigentümer-Bedingungsparameter, die in Anfragen für diese Vorgänge enthalten sind
- Die Bucket-Eigentümer-Bedingung überprüft nur, dass das im Verifizierungsparameter angegebene Konto den Bucket besitzt. Die Bucket-Eigentümer-Bedingung überprüft nicht die Konfiguration des Buckets. Sie garantiert auch nicht, dass die Konfiguration des Buckets bestimmte Bedingungen erfüllt oder mit einem früheren Zustand übereinstimmt.

Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen wurden, und um [Zugriffskontrolllisten zu deaktivieren oder zu aktivieren \(\) ACLs](#). Standardmäßig ist für Object Ownership die erzwungene Einstellung Bucket Owner festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

In den meisten modernen Anwendungsfällen in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs, und wir empfehlen, diese ACLs Option weiterhin zu deaktivieren, außer in ungewöhnlichen Fällen, in denen Sie den Zugriff für jedes Objekt einzeln kontrollieren müssen. Wenn diese ACLs Option deaktiviert ist, können Sie Richtlinien verwenden, um den Zugriff auf jedes Objekt in Ihrem Bucket einfacher zu kontrollieren, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat.

Object Ownership verfügt über drei Einstellungen, mit denen Sie die Inhaberschaft der in Ihren Bucket hochgeladenen Objekte kontrollieren und deaktivieren oder aktivieren können ACLs:

ACLsdeaktiviert

- Bucket Owner enforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr

auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLsaktiviert

- **Bevorzugter Bucket-Besitzer** — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- **Objekt-Writer** — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Für die meisten modernen Anwendungsfälle in S3 empfehlen wir, die ACLs Deaktivierung beizubehalten, indem Sie die Einstellung `Bucket Owner enforced` anwenden und Ihre Bucket-Richtlinie verwenden, um Daten nach Bedarf mit Benutzern außerhalb Ihres Kontos zu teilen. Dieser Ansatz vereinfacht die Berechtigungsverwaltung. Sie können die Option sowohl für neu erstellte als auch für bereits bestehende Buckets deaktivierenACLs. Bei neu erstellten Buckets ACLs sind diese standardmäßig deaktiviert. Bei einem vorhandenen Bucket, der bereits Objekte enthält, ACLs sind das Objekt und der Bucket nach der Deaktivierung ACLs nicht mehr Teil einer Zugriffsevaluierung, und der Zugriff wird auf der Grundlage von Richtlinien gewährt oder verweigert. Bestehende Buckets können Sie ACLs jederzeit wieder aktivieren, nachdem Sie sie deaktiviert haben. Ihr bereits vorhandener Bucket und Ihr Objekt ACLs werden wiederhergestellt.

Wir empfehlen IhnenACLs, vor der Deaktivierung Ihre Bucket-Richtlinie zu überprüfen, um sicherzustellen, dass sie alle Möglichkeiten abdeckt, mit denen Sie außerhalb Ihres Kontos Zugriff auf Ihren Bucket gewähren möchten. Nach der Deaktivierung ACLs akzeptiert Ihr Bucket nur PUT Anfragen, die kein ACL oder PUT Anfragen angeben, bei denen der Bucket-Besitzer die volle Kontrolle hatACLs, z. B. `bucket-owner-full-control` vorgefertigte ACL oder gleichwertige Formen davon, ACL ausgedrückt inXML. Bestehende Anwendungen, die die Vollzugriff ACLs des Bucket-Besitzers unterstützen, haben keine Auswirkungen. PUTAnfragen, die andere beinhalten ACLs (z. B. benutzerdefinierte Zuschüsse für bestimmte AWS-Konten) schlagen fehl und geben einen `400` Fehler mit dem Fehlercode zurück`AccessControlListNotSupported`.

Im Gegensatz dazu akzeptiert und berücksichtigt ein Bucket mit der bevorzugten Einstellung „Bucket-Besitzer“ weiterhin Bucket und ObjektACLs. Mit dieser Einstellung gehören neue Objekte, die mit der `bucket-owner-full-control` gespeicherten Option geschrieben wurden, automatisch dem Bucket-Besitzer und nicht dem Object-Writer. ACL Alle anderen ACL Verhaltensweisen bleiben bestehen. Um zu verlangen, dass alle Amazon S3 PUT S3-Operationen die gespeicherten `bucket-`

owner-full-control Daten enthalten ACL, können Sie [eine Bucket-Richtlinie hinzufügen](#), die nur Objekt-Uploads mit dieser ACL Methode zulässt.

Wenn Sie feststellen möchten, welche Einstellungen für die Objekteigentümerschaft auf Ihre Buckets angewendet werden, können Sie die Metriken von Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter [Verwenden von S3 Storage Lens, um Einstellungen für die Objekteigentümerschaft zu finden](#).

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Einstellungen für Object Ownership

Diese Tabelle zeigt die Auswirkungen, die jede Einstellung „Objekteigentum“ auf Objekte ACLs, Objekteigentum und Objekt-Uploads hat.

Einstellung	Gilt für	Auswirkung auf Object Ownership	Auswirkung auf ACLs	Hochladen akzeptiert
„Bucket-Eigentümer erzwungen“ (Standard)	Alle neuen und bestehenden Objekte	Bucket-Eigentümer besitzt jedes Objekt.	ACLs sind deaktiviert und wirken sich nicht mehr auf die Zugriffsberechtigungen für Ihren Bucket aus. Anfragen zum Einrichten oder Aktualisieren ACLs schlagen fehl. Leseanfragen	Uploads, bei denen der Bucket-Besitzer die volle Kontrolle hat, ACLs oder Uploads, bei denen kein ACL

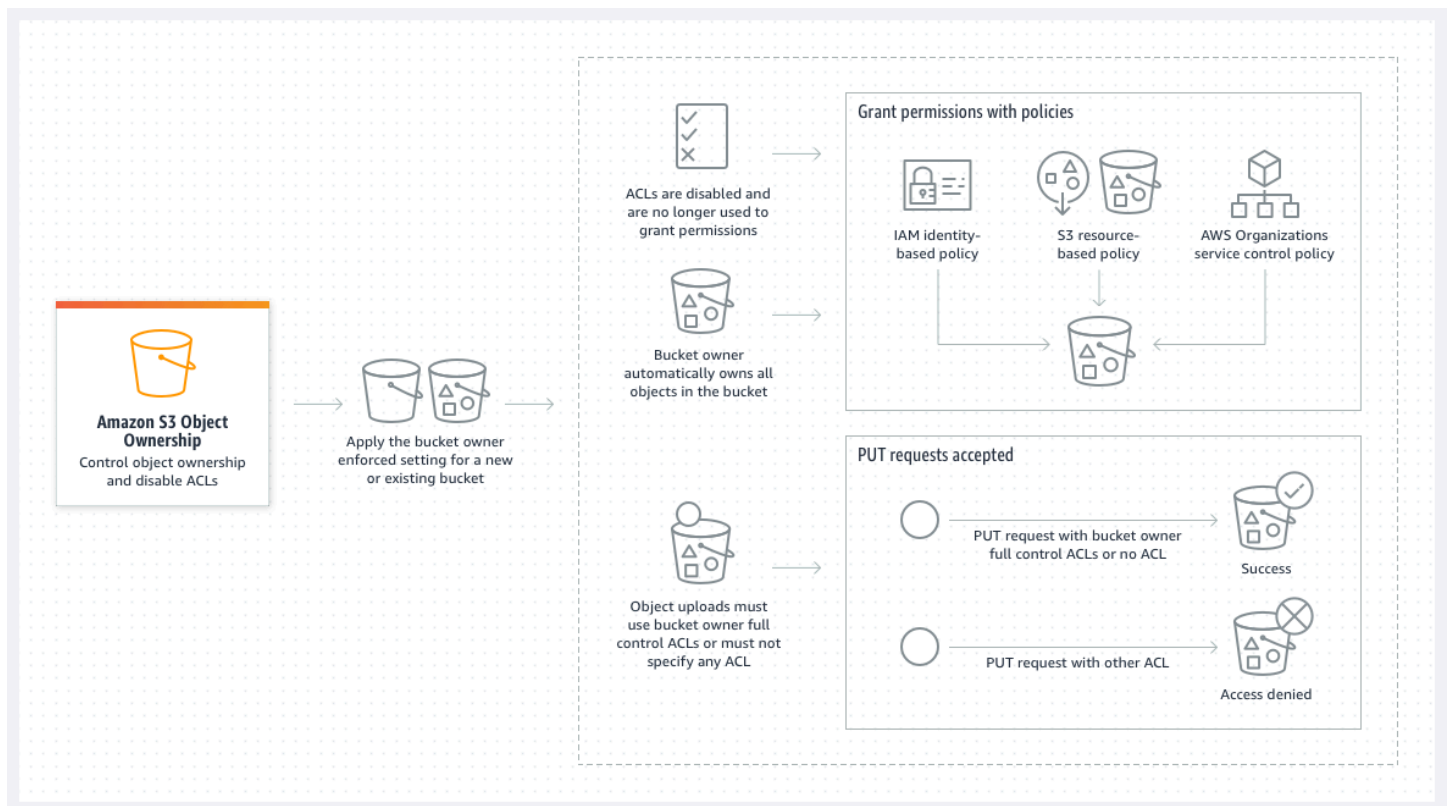
Einstellung	Gilt für	Auswirkung auf Object Ownership	Auswirkung auf ACLs	Hochladen akzeptiert
			<p>ACLs werden jedoch unterstützt.</p> <p>Bucket-Eigentümer hat das volle Eigentum und die volle Kontrolle.</p> <p>Der Objekt-Writer hat nicht mehr das volle Eigentum und die volle Kontrolle.</p>	

Einstellung	Gilt für	Auswirkung auf Object Ownership	Auswirkung auf ACLs	Hochladen akzeptiert
Bucket-Eigentümer bevorzugt	Neue Objekte	<p>Wenn bei einem Objekt-Upload auch das gespeicherte bucket-owner-full-control Objekt enthalten istACL, gehört das Objekt dem Bucket-Besitzer.</p> <p>Objekte, die zusammen mit anderen hochgeladen wurden, ACLs gehören dem Schreibkonto.</p>	<p>ACLskann aktualisiert werden und kann Berechtigungen gewähren.</p> <p>Wenn ein Objekt-Upload die bucket-owner-full-control gespeicherten Objekte enthältACL, hat der Bucket-Besitzer Vollzugriff, und der Objekt-Writer hat keinen Vollzugriff mehr.</p>	Alle Uploads
Objektschreiber	Neue Objekte	Der Objekt-Writer besitzt das Objekt.	<p>ACLskann aktualisiert werden und kann Berechtigungen gewähren.</p> <p>Der Objekt-Writer hat vollen Kontrollzugriff.</p>	Alle Uploads

Änderungen, die durch Deaktivierung eingeführt wurden ACLs

Wenn die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum angewendet wird, ACLs sind sie deaktiviert und Sie besitzen automatisch jedes Objekt im Bucket und übernehmen die volle Kontrolle über jedes Objekt im Bucket, ohne weitere Maßnahmen ergreifen zu müssen. „Bucket-Eigentümer erzwungen“ ist die Standardeinstellung für alle neu erstellten Buckets. Nachdem die Einstellung „Bucket-Eigentümer erzwungen“ angewendet wurde, sehen Sie drei Änderungen:

- Alle Buckets ACLs und Objekte ACLs sind deaktiviert, sodass Sie als Bucket-Besitzer vollen Zugriff haben. Wenn Sie eine ACL Leseanforderung für Ihren Bucket oder Ihr Objekt ausführen, werden Sie feststellen, dass nur der Bucket-Besitzer vollen Zugriff hat.
- Sie als Bucket-Eigentümer besitzen automatisch jedes Objekt in Ihrem Bucket und haben die volle Kontrolle darüber.
- ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Ihren Bucket aus. Daher basiert die Zugriffskontrolle für Ihre Daten auf Richtlinien wie Richtlinien, IAM S3-Bucket-Richtlinien, VPC Endpunktrichtlinien und OrganizationsSCPs.



Wenn Sie S3-Versioning verwenden, besitzt der Bucket-Eigentümer alle Objektversionen in Ihrem Bucket und hat die volle Kontrolle über sie. Durch das Anwenden der Einstellung „Bucket-Eigentümer erzwungen“ wird keine neue Version eines Objekts hinzugefügt.

Neue Objekte können nur dann in Ihren Bucket hochgeladen werden, wenn für sie die Vollzugriff des Bucket-Besitzers verwendet wird ACLs oder wenn kein Objekt angegeben istACL. Objekt-Uploads schlagen fehl, wenn sie ein anderes ACL angeben. Weitere Informationen finden Sie unter [Fehlerbehebung](#).

Weil die folgende PutObject Beispieloperation mit dem AWS Command Line Interface (AWS CLI) enthält die gespeicherten Objekte bucket-owner-full-controlACL, sodass das Objekt in einen Bucket mit deaktivierter Option hochgeladen werden kannACLs.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file --acl bucket-owner-full-control
```

Da der folgende PutObject Vorgang keinen spezifiziertACL, ist er auch für einen Bucket mit deaktivierter Option ACLs erfolgreich.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file
```

Note

Wenn andere AWS-Konten Wenn Sie nach dem Hochladen Zugriff auf Objekte benötigen, müssen Sie diesen Konten über Bucket-Richtlinien zusätzliche Berechtigungen gewähren. Weitere Informationen finden Sie unter [Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten](#).

Erneut aktivieren ACLs

Sie können die Aktivierung jederzeit wieder aktivieren, ACLs indem Sie von der Einstellung „Bucket-Besitzer erzwungen“ zu einer anderen Einstellung „Objektbesitz“ wechseln. Wenn Sie ein Objekt ACLs für die Rechteverwaltung verwendet haben, bevor Sie die Einstellung „Bucket-Besitzer erzwungen“ angewendet haben, und Sie diese ACL Objektberechtigungen nicht in Ihre Bucket-Richtlinie migriert haben, werden diese Berechtigungen nach der erneuten Aktivierung wiederhergestelltACLs. Darüber hinaus gehören Objekte, die in den Bucket geschrieben wurden, während die „Einstellung „Bucket-Eigentümer erzwungen“ angewendet wurde, weiterhin dem Bucket-Eigentümer.

Wenn Sie beispielsweise von der Einstellung „Bucket-Besitzer erzwungen“ zurück zur Einstellung „Objekt-Writer“ wechseln, sind Sie als Bucket-Besitzer nicht mehr Eigentümer von Objekten und haben auch keine vollständige Kontrolle mehr über Objekte, die zuvor anderen gehörten AWS-Konten. Stattdessen sind die hochladenden Konten wieder Eigentümer dieser Objekte. Objekte, die anderen Konten gehören, verwenden ACLs für Berechtigungen. Sie können also keine Richtlinien verwenden, um diesen Objekten Berechtigungen zu erteilen. Sie als Bucket-Eigentümer besitzen jedoch weiterhin alle Objekte, die in den Bucket geschrieben wurden, während die Einstellung „Bucket-Eigentümer erzwungen“ angewendet wurde. Diese Objekte gehören nicht dem Object Writer, auch wenn Sie sie erneut aktivierenACLs.

Anweisungen zur Aktivierung und Verwaltung finden ACLs Sie unter AWS Management Console, AWS Command Line Interface (CLI), RESTAPI, oder AWS SDKs, siehe [Konfiguration ACLs](#).

Voraussetzungen für die Deaktivierung ACLs

Bevor Sie die Deaktivierung ACLs für einen vorhandenen Bucket durchführen, müssen Sie die folgenden Voraussetzungen erfüllen.

Überprüfen Sie den Bucket und das Objekt ACLs und migrieren Sie die ACL Berechtigungen

Wenn Sie die Option deaktivierenACLs, wirken sich die von Bucket und Objekt erteilten Berechtigungen nicht ACLs mehr auf den Zugriff aus. Überprüfen Sie vor der Deaktivierung ACLs Ihren Bucket und Ihr ObjektACLs.

Wenn Ihr Bucket anderen Benutzern außerhalb Ihres Kontos Lese- oder Schreibberechtigungen ACLs gewährt, müssen Sie diese Berechtigungen in Ihre Bucket-Richtlinie migrieren, bevor Sie die erzwungene Einstellung für den Bucket-Besitzer anwenden können. Wenn Sie BucketsACLs, die Lese- oder Schreibzugriff außerhalb Ihres Accounts gewähren, nicht migrieren, schlägt Ihre Anfrage zur Anwendung der erzwungenen Einstellung „Bucket-Besitzer“ fehl und gibt [InvalidBucketAclWithObjectOwnership](#) Fehlercode.

Wenn Sie die Option beispielsweise ACLs für einen Bucket deaktivieren möchten, der Serverzugriffsprotokolle empfängt, müssen Sie die ACL Bucket-Berechtigungen für die S3-Protokollbereitstellungsgruppe in einer Bucket-Richtlinie auf den Logging-Serviceprinzipal migrieren. Weitere Informationen finden Sie unter [Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs](#).

Wenn Sie möchten, dass der Objektschreiber die volle Kontrolle über das hochgeladene Objekt behält, ist der Objektschreiber die beste Einstellung für die Objekteigentümerschaft in Ihrem

Anwendungsfall. Wenn Sie den Zugriff auf der Ebene einzelner Objekte steuern möchten, ist Bucket-Eigentümer bevorzugt die beste Wahl. Diese Anwendungsfälle sind ungewöhnlich.

Informationen zum Überprüfen ACLs und Migrieren von ACL Berechtigungen zu Bucket-Richtlinien finden Sie unter [Voraussetzungen für die Deaktivierung ACLs](#).

Identifizieren Sie Anfragen, ACL für die eine Autorisierung erforderlich war

Um Amazon S3 S3-Anfragen zu identifizieren, die ACLs für eine Autorisierung erforderlich sind, können Sie den `aclRequired` Wert in den Amazon S3 S3-Serverzugriffsprotokollen verwenden oder AWS CloudTrail. Wenn ACL für die Anfrage eine Autorisierung erforderlich war oder wenn Sie PUT Anfragen haben, die eine angebenACL, lautet die ZeichenfolgeYes. Wenn keine erforderlich ACLs waren oder wenn Sie einen `bucket-owner-full-control` gespeicherten Wert festlegen oder wenn die Anfragen gemäß Ihrer Bucket-Richtlinie zulässig sind, lautet die `aclRequired` Wertzeichenfolge in den Amazon S3 S3-Serverzugriffsprotokollen - "" und fehlt in CloudTrail. ACL Weitere Informationen zu den erwarteten Werten für `aclRequired` finden Sie unter [aclRequired-Werte für allgemeine Amazon-S3-Anfragen](#).

Wenn Sie `PutObjectAcl` Anfragen mit Headern haben`PutBucketAcl`, die bestimmte ACL Berechtigungen gewähren, mit Ausnahme der `bucket-owner-full-control` gespeichertenACL, müssen Sie diese Header entfernen, bevor Sie sie deaktivieren können. ACLs Andernfalls schlagen Ihre Anfragen fehl.

Für alle anderen Anfragen, ACL für die eine Autorisierung erforderlich ist, migrieren Sie diese ACL Berechtigungen in die Bucket-Richtlinien. Entfernen Sie anschließend alle Buckets, ACLs bevor Sie die erzwungene Einstellung für den Bucket-Besitzer aktivieren.

Note

Objekt ACLs nicht entfernen. Andernfalls verlieren Anwendungen, die ACLs für Berechtigungen auf Objekte angewiesen sind, den Zugriff.

Wenn Sie feststellen, dass keine Anfragen ACL zur Autorisierung erforderlich sind, können Sie mit der Deaktivierung fortfahrenACLs. Weitere Informationen zur Identifizierung von Anfragen finden Sie unter [Verwenden von Amazon-S3-Serverzugriffsprotokollen zur Identifizierung von Anforderungen](#) und [Identifizieren von Amazon S3 S3-Anfragen mit CloudTrail](#).

Überprüfen und aktualisieren Sie Bucket-Richtlinien, die verwandte ACL Bedingungsschlüssel verwenden

Nachdem Sie die Einstellung Bucket Owner enforced auf Deaktivierung angewendet haben, können neue Objekte nur dann in Ihren Bucket hochgeladen werden, wenn für die Anfrage die Vollzugriff auf den Bucket Owner verwendet wird ACLs oder wenn kein Bucket Owner angegeben ist. Überprüfen Sie vor der Deaktivierung ACLs Ihre Bucket-Richtlinie auf zugehörige ACL Bedingungsschlüssel.

Wenn in Ihrer Bucket-Richtlinie ein ACL verwandter Bedingungsschlüssel verwendet wird, `bucket-owner-full-control` der ACL vordefinierte Daten verlangt (z. B. `s3:x-amz-acl`), müssen Sie Ihre Bucket-Richtlinie nicht aktualisieren. Die folgende Bucket-Richtlinie verwendet `s3:x-amz-acl`, um zu verlangen, dass `bucket-owner-full-control` die `PutObject` Anfragen ACL für S3 gesperrt werden. Diese Richtlinie verlangt immer noch, dass der Object Writer die gespeicherten Daten `bucket-owner-full-control` spezifiziert. ACLs Deaktivierte Buckets akzeptieren dies jedoch weiterhin, sodass Anfragen weiterhin erfolgreich sind, ohne dass clientseitige Änderungen erforderlich sind.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Only allow writes to my bucket with bucket owner full control",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/ExampleUser"
        ]
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
```

Wenn Ihre Bucket-Richtlinie jedoch einen verwandten ACL Bedingungsschlüssel verwendet, der einen anderen erfordert ACL, müssen Sie diesen Bedingungsschlüssel entfernen. Diese Beispiel-Bucket-Richtlinie erfordert die `public-read` ACL für `PutObject` S3-Anfragen und muss daher vor der Deaktivierung ACLs aktualisiert werden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Only allow writes to my bucket with public read access",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/ExampleUser"
        ]
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "public-read"
        }
      }
    }
  ]
}
```

Berechtigungen für Object Ownership

Um eine Einstellung für Object Ownership für einen Bucket anzuwenden, zu aktualisieren oder zu löschen, benötigen Sie die `s3:PutBucketOwnershipControls`-Berechtigung. Um die Object-Ownership-Einstellung für einen Bucket zurückzugeben, benötigen Sie die `s3:GetBucketOwnershipControls`-Berechtigung. Weitere Informationen erhalten Sie unter [Festlegen von Object Ownership beim Erstellen eines Buckets](#) und [Anzeigen der Einstellung Object Ownership für einen S3-Bucket](#).

Deaktivierung ACLs für alle neuen Buckets

Standardmäßig werden alle neuen Buckets mit der Einstellung `Bucket Owner enforced` erstellt und sind deaktiviert. ACLs Wir empfehlen, diese Option weiterhin deaktiviert zu lassen. ACLs

In der Regel empfehlen wir, stattdessen ressourcenbasierte S3-Richtlinien (Bucket-Richtlinien und Zugriffspunktrichtlinien) oder IAM Richtlinien für die Zugriffskontrolle zu verwenden. ACLs Richtlinien stellen eine vereinfachte und flexiblere Zugriffskontrolloption dar. Mit Bucket- und Zugriffspunktrichtlinien können Sie Regeln definieren, die allgemein für alle Anfragen an Ihre Amazon-S3-Ressourcen gelten.

Replikation und Object Ownership

Wenn Sie die S3-Replikation verwenden und die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, können Sie deaktivieren ACLs (wobei die Einstellung „Bucket-Besitzer“ für Objekteigentümer erzwungen wird), um den Besitz des Replikats zu ändern AWS-Konto dem der Ziel-Bucket gehört. Diese Einstellung ahmt das Verhalten der bestehenden Besitzerüberschreibung nach, ohne dass eine `s3:ObjectOwnerOverrideToBucketOwner`-Berechtigung erforderlich ist. Alle Objekte, die mit der Einstellung „Bucket-Eigentümer erzwungen“ in den Ziel-Bucket repliziert werden, gehören dem Eigentümer des Ziel-Buckets. Weitere Informationen zur Besitzerüberschreibungsoption für Replikationskonfigurationen finden Sie unter [Ändern des Replikat-Eigentümers](#).

Einstellung von Object Ownership

Sie können mithilfe der Amazon S3 S3-Konsole eine Einstellung für den Objekteigentum anwenden. AWS CLI, AWS SDKs, Amazon S3 REST API oder AWS CloudFormation. Das Folgende REST API und AWS CLI Befehle unterstützen Objekteigentum:

REST API	AWS CLI	Beschreibung
PutBucketOwnershipControls	put-bucket-ownership-controls	Erstellt oder ändert die Einstellung Object Ownership für einen vorhandenen S3-Bucket.
CreateBucket	create-bucket	Erstellt einen Bucket mit dem <code>x-amz-object-ownership</code> -Anforderungsheader zur Angabe der Einstellung für Object Ownership.

REST API	AWS CLI	Beschreibung
GetBucketOwnershipControls	get-bucket-ownership-controls	Ruft die Einstellung Object Ownership für einen Amazon-S3-Bucket ab.
DeleteBucketOwnershipControls	delete-bucket-ownership-controls	Löscht die Einstellung für Object Ownership für einen Amazon-S3-Bucket.

Weitere Informationen zum Anwenden und Arbeiten mit Object Ownership-Einstellungen finden Sie in den folgenden Themen.

Themen

- [Voraussetzungen für die Deaktivierung ACLs](#)
- [Festlegen von Object Ownership beim Erstellen eines Buckets](#)
- [Einstellung für Object Ownership für einen vorhandenen Bucket](#)
- [Anzeigen der Einstellung Object Ownership für einen S3-Bucket](#)
- [Deaktivierung ACLs für alle neuen Buckets und Durchsetzung der Objekteigentümerschaft](#)
- [Fehlerbehebung](#)

Voraussetzungen für die Deaktivierung ACLs

Wenn Ihr ACL Bucket Zugriff außerhalb Ihres AWS-Konto, bevor Sie sie deaktivieren ACLs, müssen Sie Ihre ACL Bucket-Berechtigungen auf Ihre Bucket-Richtlinie migrieren und Ihren Bucket ACL auf die Standardeinstellung „Privat“ zurücksetzen ACL. Wenn Sie diese Buckets nicht migrieren ACLs, schlägt Ihre Anfrage ACLs fehl, die vom Bucket-Besitzer erzwungene Deaktivierung anzuwenden und gibt den [InvalidBucketAclWithObjectOwnership](#) Fehlercode. Wir empfehlen Ihnen außerdem, Ihre ACL Objektberechtigungen zu überprüfen und sie in Ihre Bucket-Richtlinie zu migrieren. Weitere Informationen zu anderen empfohlenen Voraussetzungen finden Sie unter [Voraussetzungen für die Deaktivierung ACLs](#).

Für jeden Ihrer vorhandenen Buckets und Objekte ACLs gibt es ein Äquivalent in einer IAM Richtlinie. Die folgenden Beispiele für Bucket-Richtlinien zeigen Ihnen, wie READ und wie die WRITE Berechtigungen für Bucket und Objekt ACLs den IAM Berechtigungen zugeordnet werden. Weitere

Informationen darüber, wie beide Begriffe in IAM Berechtigungen ACL übersetzt werden, finden Sie unter [Zuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen](#).

Informationen zur Überprüfung und Migration von ACL Berechtigungen zu Bucket-Richtlinien finden Sie in den folgenden Themen.

Themen

- [Beispiele für Bucket-Richtlinie](#)
- [Verwenden Sie die S3-Konsole, um ACL Berechtigungen zu überprüfen und zu migrieren](#)
- [Verwendung der AWS CLI um ACL Berechtigungen zu überprüfen und zu migrieren](#)
- [Beispielhafte Walkthroughs](#)

Beispiele für Bucket-Richtlinie

Diese Beispiel-Bucket-Richtlinien zeigen Ihnen, wie Sie WRITE Bucket READ - und ACL Objektberechtigungen für Drittanbieter migrieren AWS-Konto zu einer Bucket-Richtlinie. READ_ACP und WRITE_ACP ACLs sind für Richtlinien weniger relevant, da sie ACL verwandte Berechtigungen gewähren (s3:GetBucketAc1s3:GetObjectAc1,s3:PutBucketAc1, und s3:PutObjectAc1).

Example — **READ** ACL für einen Eimer

Wenn dein Eimer einen hätte READACL, der gewährt AWS-Konto **111122223333** Wenn Sie berechtigt sind, den Inhalt Ihres Buckets aufzulisten, können Sie eine Bucket-Richtlinie schreiben, dies3:ListBucket,s3:ListBucketVersions, s3:ListBucketMultipartUploads Berechtigungen für Ihren Bucket gewährt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Permission to list the objects in a bucket",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      }
    }
  ],
}
```

```

        "Action": [
            "s3:ListBucket",
            "s3:ListBucketVersions",
            "s3:ListBucketMultipartUploads"
        ],
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    }
]
}

```

Example — **READ** ACLs für jedes Objekt in einem Bucket

Wenn jedes Objekt in Ihrem Bucket eine hat READACL, die Zugriff gewährt auf AWS-Konto **111122223333**, können Sie eine Bucket-Richtlinie schreiben, die diesem Konto für jedes Objekt in Ihrem Bucket `s3:GetObjectVersion` Berechtigungen `s3:GetObject` und Berechtigungen gewährt.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Read permission for every object in a bucket",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    }
  ]
}

```

Dieses Beispiel-Ressourcenelement gewährt Zugriff auf ein bestimmtes Objekt.

```

"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"

```

Example — WRITE ACL das gewährt Berechtigungen zum Schreiben von Objekten in einen Bucket

Wenn Ihr Bucket einen hat WRITEACL, der gewährt AWS-Konto **111122223333** Wenn Sie berechtigt sind, Objekte in Ihren Bucket zu schreiben, können Sie eine Bucket-Richtlinie schreiben, die `s3:PutObject` Berechtigungen für Ihren Bucket erteilt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Permission to write objects to a bucket",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    }
  ]
}
```

Verwenden Sie die S3-Konsole, um ACL Berechtigungen zu überprüfen und zu migrieren

Um die ACL Berechtigungen eines Buckets zu überprüfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Überprüfen Sie unter Zugriffskontrollliste (ACL) Ihre ACL Bucket-Berechtigungen.

Um die ACL Berechtigungen eines Objekts zu überprüfen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der Ihr Objekt enthält.

3. Wählen Sie in der Liste Objekts (Objekte) den Objektnamen aus.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Überprüfen Sie unter Zugriffskontrollliste (ACL) Ihre ACL Objektberechtigungen.

Um ACL Berechtigungen zu migrieren und Ihren Bucket zu aktualisieren ACL

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus.
3. Wählen Sie auf der Registerkarte Berechtigungen unter Bucket-Richtlinie die Option Bearbeiten aus.
4. Fügen Sie im Feld Richtlinie Ihre Bucket-Richtlinie hinzu oder aktualisieren Sie sie.

Beispiele für Bucket-Richtlinien finden Sie unter [Beispiele für Bucket-Richtlinie](#) und [Beispielhafte Walkthroughs](#).

5. Wählen Sie Änderungen speichern.
6. [Aktualisieren Sie Ihren Bucket ACL](#), um ACL Zuschüsse für andere Gruppen zu entfernen, oder AWS-Konten.
7. [Wenden Sie die Einstellung Bucket-Eigentümer erzwungen](#) für die Objekteigentümerschaft an.

Verwendung der AWS CLI um ACL Berechtigungen zu überprüfen und zu migrieren

1. Um den Bucket ACL für Ihren Bucket zurückzugeben, verwenden Sie den [get-bucket-acl](#) AWS CLI Befehl:

```
aws s3api get-bucket-acl --bucket amzn-s3-demo-bucket
```

Dieser Bucket ACL gewährt WRITE beispielsweise READ Zugriff auf ein Drittanbieter-Konto. Dabei wird ACL das Drittanbieter-Konto anhand der [kanonischen Benutzer-ID](#) identifiziert. Um die Einstellung „Bucket-Besitzer erzwungen“ anzuwenden und zu deaktivieren ACLs, müssen Sie diese Berechtigungen für das Drittanbieter-Konto auf eine Bucket-Richtlinie migrieren.

```
{
  "Owner": {
    "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
    "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
  }
}
```

```

    },
    "Grants": [
      {
        "Grantee": {
          "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
          "ID":
"852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
          "Type": "CanonicalUser"
        },
        "Permission": "FULL_CONTROL"
      },
      {
        "Grantee": {
          "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
          "ID":
"72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
          "Type": "CanonicalUser"
        },
        "Permission": "READ"
      },
      {
        "Grantee": {
          "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
          "ID":
"72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
          "Type": "CanonicalUser"
        },
        "Permission": "WRITE"
      }
    ]
  }
}

```

Ein anderes Beispiel finden Sie ACLs unter [Beispielhafte Walkthroughs](#).

2. Migrieren Sie Ihre ACL Bucket-Berechtigungen zu einer Bucket-Richtlinie:

In diesem Beispiel gewährt Bucket-Richtlinien `s3:PutObject` und `s3:ListBucket` Berechtigungen für ein Drittanbieterkonto. In der Bucket-Richtlinie wird das Drittanbieter-Konto identifiziert durch AWS-Konto ID (`111122223333`).

```

aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:

```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyForCrossAccountAllowUpload",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3::DOC-EXAMPLE-BUCKET/*"
      ]
    }
  ]
}
```

Weitere Beispiele für Bucket-Richtlinien finden Sie unter [Beispiele für Bucket-Richtlinie](#) und [Beispielhafte Walkthroughs](#).

- Um das ACL für ein bestimmtes Objekt zurückzugeben, verwenden Sie den [get-object-acl](#) AWS CLI Befehl.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket --key EXAMPLE-OBJECT-KEY
```

- Migrieren Sie bei Bedarf ACL Objektberechtigungen zu Ihrer Bucket-Richtlinie.

Dieses Beispiel-Ressourcenelement gewährt Zugriff auf ein bestimmtes Objekt in einer Bucket-Richtlinie.

```
"Resource": "arn:aws:s3::amzn-s3-demo-bucket/EXAMPLE-OBJECT-KEY"
```

- Setzen Sie die ACL für Ihren Bucket auf die Standardeinstellung zurückACL.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

- [Wenden Sie die Einstellung „Bucket-Eigentümer erzwungen“](#) für die Objekteigentümerschaft an.

Beispielhafte Walkthroughs

Die folgenden Beispiele zeigen Ihnen, wie Sie ACL Berechtigungen für bestimmte Anwendungsfälle auf Bucket-Richtlinien migrieren.

Themen

- [Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs](#)
- [Öffentlichen Lesezugriff auf die Objekte in einem Bucket gewähren](#)
- [Gewähren Sie Amazon ElastiCache \(RedisOSS\) Zugriff auf Ihren S3-Bucket](#)

Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs

Wenn Sie die Einstellung Bucket Owner enforced anwenden möchten, um sie ACLs für einen Ziel-Bucket für die Serverzugriffsprotokollierung (auch als Ziel-Bucket bezeichnet) zu deaktivieren, müssen Sie die ACL Bucket-Berechtigungen für die S3-Protokollzustellungsgruppe in einer Bucket-Richtlinie auf den Logging-Serviceprinzipal (`logging.s3.amazonaws.com`) migrieren. Weitere Informationen zu Berechtigungen für die Protokollzustellung finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Dieser Bucket ACL gewährt WRITE_READ_ACP Zugriff auf die S3-Protokollbereitstellungsgruppe:

```
{
  "Owner": {
    "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
    "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
  },
  "Grants": [
    {
      "Grantee": {
        "Type": "CanonicalUser",
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
      },
      "Permission": "FULL_CONTROL"
    },
    {
      "Grantee": {
```

```

        "Type": "Group",
        "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
    },
    "Permission": "WRITE"
},
{
    "Grantee": {
        "Type": "Group",
        "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
    },
    "Permission": "READ_ACP"
}
]
}

```

Um ACL Bucket-Berechtigungen für die S3-Protokollzustellungsgruppe in einer Bucket-Richtlinie auf den Logging-Serviceprinzipal zu migrieren

1. Fügen Sie Ihrem Ziel-Bucket die folgende Bucket-Richtlinie hinzu. wobei Sie die Beispielwerte ersetzen.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://policy.json
```

```

policy.json:    {
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "S3ServerAccessLogsPolicy",
        "Effect": "Allow",
        "Principal": {
          "Service": "logging.s3.amazonaws.com"
        },
        "Action": [
          "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-LOGGING-PREFIX*",
        "Condition": {
          "ArnLike": {
            "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"
          },
          "StringEquals": {

```



```

    "aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"
  }
}
]
}

```

2. Setzen Sie den Bucket ACL für Ihren Ziel-Bucket auf die Standardeinstellung zurückACL.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. [Wenden Sie die Einstellung „Von Bucket-Besitzer erzwungen“](#) für den Objektbesitz auf Ihren Ziel-Bucket an.

Öffentlichen Lesezugriff auf die Objekte in einem Bucket gewähren

Wenn Ihr Objekt öffentlichen Lesezugriff auf alle Objekte in Ihrem Bucket ACLs gewährt, können Sie diese ACL Berechtigungen in eine Bucket-Richtlinie migrieren.

Dieses Objekt ACL gewährt öffentlichen Lesezugriff auf ein Objekt in einem Bucket:

```

{
  "Owner": {
    "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
    "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
        "Type": "CanonicalUser"
      },
      "Permission": "FULL_CONTROL"
    },
    {
      "Grantee": {
        "Type": "Group",
        "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
      },
      "Permission": "READ"
    }
  ]
}

```

```

    }
  ]
}

```

Um öffentliche ACL Leseberechtigungen auf eine Bucket-Richtlinie zu migrieren

- Um öffentlichen Lesezugriff auf alle Objekte in Ihrem Bucket zu gewähren, fügen Sie die folgende Bucket-Richtlinie hinzu und ersetzen Sie die Beispielwerte.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy
file://policy.json
```

```
policy.json:
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

Um öffentlichen Zugriff auf ein bestimmtes Objekt in einer Bucket-Richtlinie zu gewähren, verwenden Sie das folgende Format für das Resource-Element.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/OBJECT-KEY"
```

Um öffentlichen Zugriff auf alle Objekte mit einem bestimmten Präfix zu gewähren, verwenden Sie das folgende Format für das Resource-Element.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/PREFIX/*"
```

- [Wenden Sie die Einstellung „Bucket-Eigentümer erzwungen“](#) für die Objekteigentümerschaft an.

Gewähren Sie Amazon ElastiCache (RedisOSS) Zugriff auf Ihren S3-Bucket

Sie können [Ihr ElastiCache \(Redis-OSS\) Backup in einen S3-Bucket exportieren](#), sodass Sie von außen auf das Backup zugreifen können. ElastiCache Um Ihr Backup in einen S3-Bucket zu exportieren, müssen Sie ElastiCache Berechtigungen zum Kopieren eines Snapshots in den Bucket erteilen. Wenn Sie ElastiCache in einem Bucket Berechtigungen erteilt haben ACL, müssen Sie diese Berechtigungen in eine Bucket-Richtlinie migrieren, bevor Sie die Einstellung Bucket owner enforced auf Deaktivierung ACLs anwenden. Weitere Informationen finden Sie unter [ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#) im ElastiCache Amazon-Benutzerhandbuch.

Das folgende Beispiel zeigt die ACL Bucket-Berechtigungen, denen Berechtigungen gewährt werden ElastiCache.

```
{
  "Owner": {
    "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
    "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
        "Type": "CanonicalUser"
      },
      "Permission": "FULL_CONTROL"
    },
    {
      "Grantee": {
        "DisplayName": "aws-scs-s3-readonly",
        "ID": "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
        "Type": "CanonicalUser"
      },
      "Permission": "READ"
    },
    {
      "Grantee": {
        "DisplayName": "aws-scs-s3-readonly",
        "ID": "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
```

```

        "Type": "CanonicalUser"
    },
    "Permission": "WRITE"
  },
  {
    "Grantee": {
      "DisplayName": "aws-scs-s3-readonly",
      "ID":
"540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
      "Type": "CanonicalUser"
    },
    "Permission": "READ_ACP"
  }
]
}

```

Um ACL Bucket-Berechtigungen für ElastiCache (RedisOSS) zu einer Bucket-Richtlinie zu migrieren

1. Fügen Sie Ihrem Bucket die folgende Bucket-Richtlinie hinzu und ersetzen Sie die Beispielwerte.

```

aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy
file://policy.json

policy.json:
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
      "Principal": {
        "Service": "Region.elasticache-snapshot.amazonaws.com"
      },
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetBucketAcl",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",

```

```
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
}
]
```

2. Setzen Sie die ACL für Ihren Bucket auf die Standardeinstellung zurückACL:

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. [Wenden Sie die Einstellung „Bucket-Eigentümer erzwungen“](#) für die Objekteigentümerschaft an.

Festlegen von Object Ownership beim Erstellen eines Buckets

Wenn Sie einen Bucket erstellen, können Sie S3 Object Ownership konfigurieren. Informationen zum Festlegen von Object Ownership für einen vorhandenen Bucket finden Sie unter [Einstellung für Object Ownership für einen vorhandenen Bucket](#).

S3 Object Ownership ist eine Einstellung auf Amazon S3-Bucket-Ebene, mit der Sie [Zugriffskontrolllisten \(ACLs\)](#) deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket übernehmen können, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. Standardmäßig ist für S3 Object Ownership die erzwungene Einstellung Bucket Owner festgelegt und ACLs für neue Buckets deaktiviert. Bei ACLs deaktivierter Option besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen.

Object Ownership verfügt über drei Einstellungen, mit denen Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren könnenACLs:

ACLsdeaktiviert

- Bucket Owner enforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLsaktiviert

- **Bevorzugter Bucket-Besitzer** — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- **Objekt-Writer** — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Berechtigungen: Um die Einstellung Bucket-Eigentümer erzwungen oder Bucket-Eigentümer bevorzugt anzuwenden, müssen Sie über die folgenden Berechtigungen verfügen: `s3:CreateBucket` und `s3:PutBucketOwnershipControls`. Wenn Sie einen Bucket mit aktivierter Einstellung Object writer (Objektschreiber), sind keine zusätzlichen Berechtigungen erforderlich. Weitere Informationen zu Amazon S3-Berechtigungen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.


Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Important

In den meisten modernen Anwendungsfällen in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs, und wir empfehlen, diese zu deaktivieren, ACLs außer in ungewöhnlichen Fällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Mit Object Ownership können Sie Richtlinien für die Zugriffskontrolle deaktivieren ACLs und sich auf diese verlassen. Wenn Sie die Option deaktivierenACLs, können Sie ganz einfach einen Bucket mit Objekten verwalten, die von verschiedenen Seiten hochgeladen wurden AWS Konten. Sie als Bucket-Eigentümer besitzen alle Objekte im Bucket und können den Zugriff darauf mithilfe von Richtlinien verwalten.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

 Note

Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Für eine Liste von Amazon S3 AWS-Regionen, siehe [AWS-Service Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.


Anschließend wird die Seite Bucket erstellen geöffnet.

5. Sehen Sie sich unter Allgemeine Konfiguration die AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Allgemein aus.
7. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

Der Bucket-Name ...:


- Muss innerhalb einer Partition einzigartig sein. Eine Partition ist eine Gruppe von Regionen. AWS hat derzeit drei Partitionen: aws (Standardregionen), aws-cn (Regionen in China) und aws-us-gov (AWS GovCloud (US) Regions).
- zwischen 3 und 63 Zeichen lang sein,
- Darf nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen. Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Das Tool AWS-Konto wer den Bucket erstellt, besitzt ihn. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

 **Important**

Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name ist in dem Bereich sichtbarURLs, der auf die Objekte im Bucket verweist.

8. AWS Management Console ermöglicht es Ihnen, die Einstellungen eines vorhandenen Buckets in Ihren neuen Bucket zu kopieren. Wenn Sie die Einstellungen eines vorhandenen Buckets nicht kopieren möchten, fahren Sie mit dem nächsten Schritt fort.

 **Note**

Diese Option:

- Ist nicht verfügbar in AWS CLI und ist nur in der Konsole verfügbar
- Ist für Verzeichnis-Buckets nicht verfügbar
- Kopiert die Bucket-Richtlinie nicht aus dem vorhandenen Bucket in den neuen Bucket

Um die Einstellungen eines vorhandenen Buckets zu kopieren, wählen Sie unter Einstellungen aus vorhandenem Bucket kopieren die Option Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geöffnet. Suchen Sie den Bucket mit den Einstellungen, die Sie kopieren möchten, und wählen Sie Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geschlossen, und das Fenster „Bucket erstellen“ wird erneut geöffnet.

Unter Einstellungen aus vorhandenem Bucket kopieren wird nun der Name des ausgewählten Buckets angezeigt. Außerdem wird die Option Standardwerte wiederherstellen angezeigt, mit der Sie die kopierten Bucket-Einstellungen entfernen können. Überprüfen Sie die verbleibenden Bucket-Einstellungen auf der Seite Bucket erstellen. Sie werden sehen, dass sie jetzt mit den Einstellungen des von Ihnen ausgewählten Buckets übereinstimmen. Sie können mit dem letzten Schritt fortfahren.

9. Wählen Sie unter Object Ownership eine der folgenden Einstellungen aus, um die Inhaberschaft von Objekten, die in Ihren Bucket hochgeladen wurden, zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

ACLsdeaktiviert

- Bucket Owner enforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

Standardmäßig ACLs sind sie deaktiviert. Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option weiterhin zu deaktivieren, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.

Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control`ACL, können Sie [eine Bucket-Richtlinie hinzufügen](#), die nur Objekt-Uploads zulässt, die dies verwenden. ACL


- Object Writer — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Note

Die Standardeinstellung ist Bucket-Eigentümer erzwungen. Um die Standardeinstellung anzuwenden und ACLs deaktiviert zu lassen, ist nur die `s3:CreateBucket` entsprechende Berechtigung erforderlich. Zum Aktivieren ACLs benötigen Sie die `s3:PutBucketOwnershipControls` entsprechende Genehmigung.

10. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.

Alle vier Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

 Note

Zum Aktivieren aller Einstellungen zum Blockieren des öffentlichen Zugriffs ist nur die `s3:CreateBucket`-Berechtigung erforderlich. Wenn Sie eine der Einstellungen zum Blockieren des öffentlichen Zugriffs deaktivieren möchten, benötigen Sie die `s3:PutBucketPublicAccessBlock`-Berechtigung.

11. (Optional) Unter Bucket Versioning (Bucket-Versionsverwaltung) können Sie auswählen, ob Sie Varianten von Objekten in Ihrem Bucket beibehalten möchten. Weitere Informationen über das Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).


Wenn Sie die Versionsverwaltung in Ihrem Bucket deaktivieren oder aktivieren möchten, wählen Sie entweder Disable (Deaktivieren) oder Enable (Aktivieren) aus.

12. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher verwendet werden.

Wenn Sie ein Bucket-Tag hinzuzufügen, geben Sie einen Key (Schlüssel) und optional einen Value (Wert) ein. Wählen Sie dann Add Tag (Tag hinzufügen) aus.

13. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
14. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Standardverschlüsselung zu konfigurieren:

- Verwalteter Amazon S3 S3-Schlüssel (SSE-S3)
- AWS Key Management Service Schlüssel (SSE-KMS)

 Important

Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, unterliegen Sie dem Kontingent für Anfragen pro Sekunde (RPS) von AWS KMS. Für weitere Informationen über AWS

KMS Kontingente und wie Sie eine Quotenerhöhung beantragen können, finden Sie unter [Kontingente](#) im AWS Key Management Service Leitfaden für Entwickler.

Buckets und neue Objekte werden mit serverseitiger Verschlüsselung verschlüsselt. Dabei ist ein Von Amazon S3 verwalteter Schlüssel die Grundebene der Verschlüsselungskonfiguration. Weitere Informationen zur Standardverschlüsselung finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

15. Wenn du wählst AWS Key Management Service Taste (SSE-KMS), gehen Sie wie folgt vor:
 - a. Unter AWS KMS key, geben Sie Ihren KMS Schlüssel auf eine der folgenden Arten an:
 - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie Ihren KMS Schlüssel aus der Liste der verfügbaren Schlüssel aus.
- Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.
- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN, und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
 - Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) im AWS Key Management Service Leitfaden für Entwickler.

 **Important**

Sie können nur KMS Schlüssel verwenden, die in derselben Version verfügbar sind AWS-Region wie der Eimer. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS

Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie unter [KMSSchlüssel erstellen, die andere Konten verwenden können](#), in der AWS Key Management Service Leitfaden für Entwickler. Weitere Informationen zu SSE - finden KMS Sie unter [Geben Sie die serverseitige Verschlüsselung an mit AWS KMS \(SSE-KMS\)](#).

Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter [Identifizieren symmetrischer und KMS asymmetrischer Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.


Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) im AWS Key Management Service Leitfaden für Entwickler. Weitere Informationen zur Verwendung von AWS KMS mit Amazon S3, siehe [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

- b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren. S3-Bucket Keys senken die Verschlüsselungskosten, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).

Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

16. (Optional) Wenn Sie die S3-Objektsperre aktivieren möchten, gehen Sie wie folgt vor:

- a. Wählen Sie Erweiterte Einstellungen aus.

 **Important**

Durch Aktivieren der Objektsperre wird auch die Versioning für den Bucket aktiviert. Nach dem Aktivieren müssen Sie die Standardeinstellungen für die Objektsperre im

Hinblick auf die (rechtliche) Aufbewahrung konfigurieren, um neue Objekte vor dem Löschen oder Überschreiben zu schützen.

- b. Wenn Sie die Objektsperre aktivieren möchten, wählen Sie Enable (Aktivieren) aus, lesen Sie die angezeigte Warnung und bestätigen Sie sie.

Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Note

Wenn Sie einen Bucket mit aktivierter Objektsperre erstellen möchten, benötigen Sie die folgenden Berechtigungen: `s3:CreateBucket`, `s3:PutBucketVersioning` und `s3:PutBucketObjectLockConfiguration`.

17. Wählen Sie Bucket erstellen aus.

Verwendung der AWS CLI

Um bei der Erstellung eines neuen Buckets den Objekteigentum festzulegen, verwenden Sie `create-bucket` AWS CLI Befehl mit dem `--object-ownership` Parameter.

In diesem Beispiel wird die vom Bucket Owner erzwungene Einstellung für einen neuen Bucket angewendet, wobei AWS CLI:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --region us-east-1 --object-ownership BucketOwnerEnforced
```

Important

Wenn Sie beim Erstellen eines Buckets mithilfe von AWS CLI, ist die Standardeinstellung `ObjectWriter` (ACLsaktiviert).

Verwendung der AWS SDK für Java

In diesem Beispiel wird die erzwungene Einstellung des Bucket-Besitzers für einen neuen Bucket festgelegt, und zwar mit dem AWS SDK for Java:

```
// Build the ObjectOwnership for CreateBucket
```

```
CreateBucketRequest createBucketRequest = CreateBucketRequest.builder()
    .bucket(bucketName)
    .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
    .build()

// Send the request to Amazon S3
s3client.createBucket(createBucketRequest);
```

Die Verwendung von AWS CloudFormation

So verwenden Sie die `AWS::S3::Bucket` AWS CloudFormation Ressource zur Festlegung des Objektbesitzes, wenn Sie einen neuen Bucket erstellen, siehe [OwnershipControls innerhalb AWS::S3::Bucket](#) in der AWS CloudFormation Benutzerleitfaden.

Mit dem REST API

Um die erzwungene Einstellung des Bucket-Besitzers für S3 Object Ownership anzuwenden, verwenden Sie den `CreateBucket` API Vorgang mit der Einstellung des `x-amz-object-ownership` Anforderungsheaders auf `BucketOwnerEnforced`. Informationen und Beispiele finden Sie unter [CreateBucket](#) in der Amazon Simple Storage Service API Reference.

Nächste Schritte: Nach der Anwendung der Einstellungen „Von Bucket-Besitzer erzwungen“ oder „Von Bucket-Besitzer bevorzugt“ auf den Objektbesitz können Sie die folgenden Schritte ausführen:

- [Bucket-Besitzer erzwungen — Erfordert](#), dass alle neuen Buckets mit ACLs deaktivierter Einstellung erstellt werden, indem eine Richtlinie IAM oder Organizations verwendet wird.
- [Bucket-Besitzer bevorzugt](#) — Fügen Sie eine S3-Bucket-Richtlinie hinzu, die vorschreibt, dass ACL für alle Objekt-Uploads `bucket-owner-full-control` in Ihren Bucket die Option „Gespeichert“ gilt.

Einstellung für Object Ownership für einen vorhandenen Bucket

Sie können S3 Object Ownership für einen vorhandenen S3-Bucket konfigurieren. Zum Anwenden von Objekt-Ownership beim Erstellen eines Buckets finden Sie unter [Festlegen von Object Ownership beim Erstellen eines Buckets](#).

S3 Object Ownership ist eine Einstellung auf Amazon S3-Bucket-Ebene, mit der Sie [Zugriffskontrolllisten \(ACLs\)](#) deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket übernehmen können, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten

vereinfacht wird. Standardmäßig ist für S3 Object Ownership die erzwungene Einstellung Bucket Owner festgelegt und ACLs für neue Buckets deaktiviert. Bei ACLs deaktivierter Option besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen.

Object Ownership verfügt über drei Einstellungen, mit denen Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren können ACLs:

ACLs deaktiviert

- Bucket Owner enforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLs wirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLs aktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreiben ACL.
- Objekt-Writer — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Voraussetzungen: Bevor Sie die Deaktivierung der Einstellung Bucket Owner enforced anwenden ACLs, müssen Sie ACL Bucket-Berechtigungen auf Bucket-Richtlinien migrieren und Ihren Bucket auf die Standardeinstellung „ACLs Privat“ zurücksetzen. ACL Wir empfehlen Ihnen außerdem, ACL Objektberechtigungen auf Bucket-Richtlinien zu migrieren und Bucket-Richtlinien zu bearbeiten, für die ACLs andere Benutzer als den Bucket-Besitzer Vollzugriff ACLs erforderlich sind. Weitere Informationen finden Sie unter [Voraussetzungen für die Deaktivierung ACLs](#).

Berechtigungen: Um diesen Vorgang zu verwenden, müssen Sie die `s3:PutBucketOwnershipControls`-Berechtigung haben. Weitere Informationen zu Amazon S3-Berechtigungen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Verwenden der S3-Konsole

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, auf den Sie eine Einstellung für S3 Object Ownership anwenden möchten.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Wählen Sie unter Object Ownership die Option Edit (Bearbeiten).
5. Wählen Sie unter Object Ownership eine der folgenden Einstellungen, um die Inhaberschaft der in Ihren Bucket hochgeladenen Objekte zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

ACLsdeaktiviert

- Bucket-Besitzer erzwungen — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle über jedes Objekt im Bucket. ACLswirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

Um zu verlangen, dass alle neuen Buckets mit ACLs deaktivierter Option erstellt werden, indem Sie oder verwenden IAM AWS Organizations Richtlinien finden Sie unter [Deaktivierung ACLs für alle neuen Buckets \(Bucket Owner wird erzwungen\)](#).

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.

Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control`ACL, können Sie [eine Bucket-Richtlinie hinzufügen](#), die nur Objekt-Uploads zulässt, die dies verwenden. ACL

- Object Writer — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs
6. Wählen Sie Save (Speichern) aus.

Verwendung der AWS CLI

Um eine Einstellungen für Object Ownership für einen vorhandenen Bucket anzuwenden, verwenden Sie den `put-bucket-ownership-controls`-Befehl mit dem `--ownership-controls`-Parameter. Gültige Werte für die Eigentümerschaft sind `BucketOwnerEnforced`, `BucketOwnerPreferred` oder `ObjectWriter`.

In diesem Beispiel wird die erzwungene Einstellung „Bucket Owner“ auf einen vorhandenen Bucket angewendet, indem AWS CLI:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Weitere Informationen zu `put-bucket-ownership-controls` finden Sie unter [.put-bucket-ownership-controls](#) in der AWS Command Line Interface Benutzerleitfaden.

Verwendung der AWS SDK für Java

In diesem Beispiel wird die `BucketOwnerEnforced` Einstellung für Object Ownership auf einen vorhandenen Bucket angewendet, indem AWS SDK for Java:

```
// Build the ObjectOwnership for BucketOwnerEnforced
OwnershipControlsRule rule = OwnershipControlsRule.builder()
    .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
    .build();

OwnershipControls ownershipControls = OwnershipControls.builder()
    .rules(rule)
    .build()

// Build the PutBucketOwnershipControlsRequest
PutBucketOwnershipControlsRequest putBucketOwnershipControlsRequest =
    PutBucketOwnershipControlsRequest.builder()
        .bucket(BUCKET_NAME)
        .ownershipControls(ownershipControls)
        .build();

// Send the request to Amazon S3
s3client.putBucketOwnershipControls(putBucketOwnershipControlsRequest);
```

Die Verwendung von AWS CloudFormation

Zur Verwendung AWS CloudFormation Informationen zum Anwenden der Einstellung „Objekteigentum“ auf einen vorhandenen Bucket finden Sie unter [AWS::S3::Bucket OwnershipControls](#) in der AWS CloudFormation Benutzerleitfaden.

Mit dem REST API

Um die Einstellung REST API Objekteigentum auf einen vorhandenen S3-Bucket anzuwenden, verwenden Sie `PutBucketOwnershipControls`. Weitere Informationen finden Sie unter [PutBucketOwnershipControls](#) in der Amazon Simple Storage Service API Reference.

Nächste Schritte: Nach der Anwendung der Einstellungen „Von Bucket-Besitzer erzwungen“ oder „Von Bucket-Besitzer bevorzugt“ auf den Objektbesitz können Sie die folgenden Schritte ausführen:

- [Bucket-Besitzer erzwungen — Erfordert](#), dass alle neuen Buckets mit ACLs deaktivierter Einstellung erstellt werden, indem eine Richtlinie IAM oder Organizations verwendet wird.
- [Bucket-Besitzer bevorzugt](#) — Fügen Sie eine S3-Bucket-Richtlinie hinzu, die vorschreibt, dass ACL für alle Objekt-Uploads `bucket-owner-full-control` in Ihren Bucket die Option „Gespeichert“ gilt.

Anzeigen der Einstellung Object Ownership für einen S3-Bucket

S3 Object Ownership ist eine Einstellung auf Amazon S3-Bucket-Ebene, mit der Sie [Zugriffskontrolllisten \(ACLs\)](#) deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket übernehmen können, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. Standardmäßig ist für S3 Object Ownership die erzwungene Einstellung Bucket Owner festgelegt und ACLs für neue Buckets deaktiviert. Bei ACLs deaktivierter Option besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen.

Object Ownership verfügt über drei Einstellungen, mit denen Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren können ACLs:

ACLs deaktiviert

- Bucket Owner enforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLs wirkt sich nicht mehr

auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLsaktiviert

- **Bevorzugter Bucket-Besitzer** — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- **Objekt-Writer** — Der AWS-Konto derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Sie können die S3 Object Ownership-Einstellungen für einen Amazon-S3-Bucket anzeigen. Informationen zum Festlegen von Object Ownership für einen neuen Bucket finden Sie unter [Festlegen von Object Ownership beim Erstellen eines Buckets](#). Informationen zum Festlegen von Object Ownership für einen vorhandenen Bucket finden Sie unter [Einstellung für Object Ownership für einen vorhandenen Bucket](#).

Berechtigungen Um diesen Vorgang verwenden zu können, müssen Sie die `s3:GetBucketOwnershipControls`-Berechtigung haben. Weitere Informationen zu Amazon S3-Berechtigungen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, auf den Sie eine Einstellung für Object Ownership anwenden möchten.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Unter Object Ownership können Sie die Objekteigentümerschafts-Einstellungen für Ihren Bucket anzeigen.

Verwendung der AWS CLI

Um die Einstellung S3 Object Ownership für einen S3-Bucket abzurufen, verwenden Sie [get-bucket-ownership-controls](#) AWS CLI Befehl.

```
aws s3api get-bucket-ownership-controls --bucket amzn-s3-demo-bucket
```

Mit dem REST API

Verwenden Sie den `GetBucketOwnershipControls` API Vorgang, um die Einstellung Object Ownership für einen S3-Bucket abzurufen. Weitere Informationen finden Sie unter [GetBucketOwnershipControls](#).

Deaktivierung ACLs für alle neuen Buckets und Durchsetzung der Objekteigentümerschaft

Wir empfehlen, dass Sie die Option in Ihren ACLs Amazon S3 S3-Buckets deaktivieren. Wenden Sie dazu die Einstellung „Bucket-Eigentümer erzwungen“ für S3 Object Ownership an. Wenn Sie diese Einstellung anwenden, ACLs sind sie deaktiviert und Sie besitzen automatisch alle Objekte in Ihrem Bucket und haben die volle Kontrolle über sie. Um zu verlangen, dass alle neuen Buckets mit ACLs deaktivierter Einstellung erstellt werden, verwenden Sie AWS Identity and Access Management (IAM) Richtlinien oder AWS Organizations Richtlinien zur Dienstkontrolle (SCPs), wie im nächsten Abschnitt beschrieben.

Um den Objekteigentum für neue Objekte ohne Deaktivierung durchzusetzenACLs, können Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden. Wenn Sie diese Einstellung anwenden, empfehlen wir Ihnen dringend, Ihre Bucket-Richtlinie dahingehend zu aktualisieren, dass ACL für alle PUT Anfragen an Ihren Bucket „gesperrt“ gelten muss. `bucket-owner-full-control` Stellen Sie sicher, dass Sie auch Ihre Kunden so aktualisieren, dass sie die `bucket-owner-full-control` gescannten Dateien von anderen Konten ACL an Ihren Bucket senden.

Themen

- [Deaktivierung ACLs für alle neuen Buckets \(Bucket Owner wird erzwungen\)](#)
- [Die bucket-owner-full-control ACL für Amazon S3 gespeicherten PUT Operationen werden benötigt \(Bucket-Besitzer bevorzugt\)](#)

Deaktivierung ACLs für alle neuen Buckets (Bucket Owner wird erzwungen)

Die folgende IAM Beispielrichtlinie verweigert einem bestimmten IAM Benutzer oder einer bestimmten Rolle die `s3:CreateBucket` Berechtigung, sofern nicht die erzwungene Einstellung des Bucket-Besitzers für Objekteigentümer angewendet wird. Das Schlüssel-Wert-Paar im Condition-Block gibt `s3:x-amz-object-ownership` als Schlüssel und die `BucketOwnerEnforced`-Einstellung als seinen Wert an. Mit anderen Worten, der IAM Benutzer kann Buckets nur erstellen, wenn er die erzwungene Einstellung „Bucket Ownership“ auf Object Ownership gesetzt und deaktiviert hat. ACLs Sie können diese Richtlinie auch als Grenze SCP für Ihre AWS Organisation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireBucketOwnerFullControl",
      "Action": "s3:CreateBucket",
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-object-ownership": "BucketOwnerEnforced"
        }
      }
    }
  ]
}
```

Die `bucket-owner-full-control` ACL für Amazon S3 gespeicherten **PUT** Operationen werden benötigt (Bucket-Besitzer bevorzugt)

Mit der bevorzugten Einstellung „Bucket Ownership“ für Object Ownership besitzen Sie als Bucket-Besitzer neue Objekte, die andere Konten zusammen mit den gespeicherten Objekten in Ihren Bucket schreiben, und haben die volle Kontrolle `bucket-owner-full-control` darüberACL. Wenn jedoch andere Konten Objekte in Ihren Bucket schreiben, ohne dass die gespeicherten Objekte gespeichert sind `bucket-owner-full-control`ACL, behält der Object Writer die volle Kontrolle über den Zugriff. Sie als Besitzer des Buckets können eine Bucket-Richtlinie implementieren, die Schreibvorgänge nur dann erlaubt, wenn der Bucket-Server spezifiziert `bucket-owner-full-control` wirdACL.

Note

Wenn Sie diese ACLs Option mit der Einstellung Bucket Owner enforced deaktiviert haben, besitzen Sie als Bucket-Besitzer automatisch alle Objekte in Ihrem Bucket und haben die volle Kontrolle über sie. Sie müssen diesen Abschnitt nicht verwenden, um Ihre Bucket-Richtlinie zu aktualisieren, um den Objekteigentümer für den Bucket-Eigentümer durchzusetzen.

Die folgende Bucket-Richtlinie spezifiziert dieses Konto *111122223333* kann Objekte hochladen auf *amzn-s3-demo-bucket* nur wenn das Objekt auf gesetzt ACL ist *bucket-owner-full-control*. Achten Sie darauf, zu ersetzen *111122223333* mit deinem Konto und *amzn-s3-demo-bucket* mit dem Namen Ihres Buckets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Only allow writes to my bucket with bucket owner full control",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/ExampleUser"
        ]
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
```

Im Folgenden finden Sie ein Beispiel für einen Kopiervorgang, der das *bucket-owner-full-control* Speichern ACL von Dateien mit dem AWS Command Line Interface (AWS CLI).

```
aws s3 cp file.txt s3://amzn-s3-demo-bucket --acl bucket-owner-full-control
```

Wenn die Bucket-Richtlinie in Kraft gesetzt wurde und der Client die gespeicherten Dateien nicht einschließt `bucket-owner-full-control`ACL, schlägt der Vorgang fehl und der Uploader erhält die folgende Fehlermeldung:

Beim Aufrufen des `PutObject` Vorgangs ist ein Fehler aufgetreten (`AccessDenied`): Zugriff verweigert.

Note

Wenn Clients nach dem Hochladen Zugriff auf Objekte benötigen, müssen Sie dem hochladenden Konto zusätzliche Berechtigungen erteilen. Informationen zum Erteilen von Zugriff auf Ihre Ressourcen für Konten finden Sie unter [Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten](#).

Fehlerbehebung

Wenn Sie die erzwungene Einstellung des Bucket-Besitzers für S3 Object Ownership anwenden, sind die Zugriffskontrolllisten (ACLs) deaktiviert und Sie als Bucket-Besitzer sind automatisch Eigentümer aller Objekte in Ihrem Bucket. ACLs wirken sich nicht mehr auf die Berechtigungen für die Objekte in Ihrem Bucket aus. Sie können Richtlinien verwenden, um Berechtigungen zu erteilen. Bei allen PUT S3-Anfragen müssen entweder die `bucket-owner-full-control` gespeicherten ACL Daten angegeben werden oder nicht. Andernfalls schlagen diese Anfragen fehl. ACL Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Wenn ein ungültiger ACL Wert angegeben ist oder ACL Bucketberechtigungen einen Zugriff außerhalb Ihres AWS-Konto, werden Ihnen möglicherweise die folgenden Fehlerantworten angezeigt.

`AccessControlListNotSupported`

Nachdem Sie die erzwungene Einstellung „Bucket-Besitzer“ für Objekteigentum angewendet haben, ACLs sind sie deaktiviert. Anfragen zum Setzen ACLs oder Aktualisieren ACLs schlagen mit einem `400` Fehler fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt. Leseanfragen geben ACLs immer eine Antwort zurück, aus der hervorgeht, dass der Bucket-Besitzer die volle Kontrolle hat. In Ihren PUT Vorgängen müssen Sie entweder den Bucket-Besitzer mit Vollzugriff angeben ACLs oder keinen ACL. Andernfalls schlagen Ihre PUT-Operationen fehl.

Das folgende Beispiel `put-object` AWS CLI Der Befehl beinhaltet den `public-read` gespeicherten Befehl `ACL`.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object-key-name --body doc-example-body --acl public-read
```

Wenn der Bucket die Einstellung `Bucket Owner enforced` zur Deaktivierung verwendet `ACLs`, schlägt dieser Vorgang fehl und der Uploader erhält die folgende Fehlermeldung:

Beim Aufrufen der `PutObject` Operation ist ein Fehler aufgetreten (`AccessControlListNotSupported`):
Der Bucket erlaubt nicht `ACLs`

`InvalidBucketAclWithObjectOwnership`

Wenn Sie die Einstellung `Bucket Owner enforced` auf Deaktivierung anwenden möchten `ACLs`, `ACL` darf Ihr Bucket nur dem Bucket-Besitzer die volle Kontrolle geben. Ihr Bucket `ACL` kann keinen Zugriff auf ein externes Objekt gewähren `AWS-Konto` oder eine andere Gruppe. Beispiel: In Ihrer `CreateBucket` Anfrage wird der `Bucket Owner enforced` festgelegt und ein Bucket spezifiziert `ACL`, das Zugriff auf eine externe `AWS-Konto`, schlägt Ihre Anfrage mit einem `400` Fehler fehl und gibt den `InvalidBucketAclWithObjectOwnership` Fehlercode zurück. Ebenso schlägt die `PutBucketOwnershipControls` Anfrage fehl, wenn Sie festlegen, dass der `Bucket Owner` für einen Bucket durchgesetzt wird `ACL`, der über einen Bucket verfügt, der anderen Berechtigungen gewährt.

Example : Ein vorhandener Bucket `ACL` gewährt öffentlichen Lesezugriff

Wenn ein vorhandener Bucket beispielsweise öffentlichen Lesezugriff `ACL` gewährt, können Sie die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum erst anwenden, wenn Sie diese `ACL` Berechtigungen in eine Bucket-Richtlinie migrieren und Ihren Bucket `ACL` auf die Standardeinstellung `privat ACL` zurücksetzen. Weitere Informationen finden Sie unter [Voraussetzungen für die Deaktivierung `ACLs`](#).

Dieser Beispiel-Bucket `ACL` gewährt öffentlichen Lesezugriff:

```
{
  "Owner": {
    "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
  },
  "Grants": [
```



```
{
  "Grantee": {
    "ID":
"852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
    "Type": "CanonicalUser"
  },
  "Permission": "FULL_CONTROL"
},
{
  "Grantee": {
    "Type": "Group",
    "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
  },
  "Permission": "READ"
}
]
```

Das folgende Beispiel `put-bucket-ownership-controls` AWS CLI Der Befehl wendet die erzwungene Einstellung `Bucket Owner` für `Object Ownership` an:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Da der Bucket öffentlichen Lesezugriff ACL gewährt, schlägt die Anfrage fehl und gibt den folgenden Fehlercode zurück:

Beim Aufrufen der `PutBucketOwnershipControls` Operation ist ein Fehler aufgetreten (`InvalidBucketAclWithObjectOwnership`): Bucket kann nicht mit `ObjectOwnership` der `BucketOwnerEnforced` Einstellung ACLs gesetzt worden sein

Protokollierung und Überwachung in Amazon S3

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon S3 und Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihres Systems sammeln AWS Lösung, damit Sie einen Mehrpunktfehler leichter debuggen können, falls einer auftritt. AWS bietet verschiedene Tools zur Überwachung Ihrer Amazon S3 S3-Ressourcen und zur Reaktion auf potenzielle Vorfälle.

Weitere Informationen finden Sie unter [Überwachen von Amazon S3](#).

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Amazon-Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein SNS Amazon-Thema gesendet oder AWS Auto Scaling Richtlinie. CloudWatch Alarme lösen keine Aktionen aus, da sie sich in einem bestimmten Zustand befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

AWS CloudTrail Logs (Protokolle)

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in Amazon S3. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon S3 gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln. Weitere Informationen finden Sie unter [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#).

Amazon GuardDuty

[Amazon GuardDuty](#) ist ein Service zur Bedrohungserkennung, der Ihre Konten, Container, Workloads und die Daten in Ihrem AWS Umgebung zur Identifizierung potenzieller Bedrohungen oder Sicherheitsrisiken für Ihre S3-Buckets. GuardDuty bietet außerdem umfangreiche

Informationen zu den erkannten Bedrohungen. GuardDuty überwacht AWS CloudTrail das Management protokolliert Bedrohungen und gibt sicherheitsrelevante Informationen preis. Dazu gehören beispielsweise Faktoren einer API Anfrage, wie der Benutzer, der die Anfrage gestellt hat, der Standort, von dem aus die Anfrage gestellt wurde, und die spezifische API Anfrage, die in Ihrer Umgebung ungewöhnlich sein könnten. GuardDuty [GuardDuty S3 Protection](#) überwacht die von allen S3-Buckets in Ihrer Umgebung gesammelten S3-Datenergebnisse CloudTrail und identifiziert potenziell anomales und böses Verhalten.

Amazon-S3-Zugriffsprotokolle

Server-Zugriffsprotokolle enthalten detaillierte Aufzeichnungen über die Anfragen, die an einen Bucket gestellt wurden. Server-Zugriffsprotokolle sind für viele Anwendungen nützlich. Beispielsweise können Zugriffsprotokoll-Informationen bei Sicherheits- und Zugriffsprüfungen nützlich sein. Weitere Informationen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

AWS Trusted Advisor

Trusted Advisor stützt sich auf bewährte Verfahren, die wir bei der Betreuung von Hunderttausenden von AWS Kunden. Trusted Advisor inspiziert deine AWS und gibt dann Empfehlungen, wenn Möglichkeiten bestehen, Geld zu sparen, die Systemverfügbarkeit und -leistung zu verbessern oder Sicherheitslücken zu schließen. Alle AWS Kunden haben Zugriff auf fünf Trusted Advisor Checks. Kunden mit einem Business- oder Enterprise-Support-Plan können sich alle ansehen Trusted Advisor Checks.

Trusted Advisor hat die folgenden Amazon S3-bezogenen Prüfungen:

- Protokollierungskonfiguration von Amazon-S3-Buckets.
- Sicherheitsprüfungen für Amazon-S3-Buckets mit offenen Zugriffsberechtigungen.
- Fehlertoleranzprüfungen für Amazon-S3-Buckets, für die keine Versionsverwaltung aktiviert oder deren Versionsverwaltung ausgesetzt ist

Weitere Informationen finden Sie unter [AWS Trusted Advisor](#) in der AWS Support Benutzerleitfaden.

Die folgenden bewährten Sicherheitsmethoden umfassen ebenfalls die Protokollierung und Überwachung:

- [Identify and audit all your Amazon S3 buckets](#)
- [Implement monitoring using Amazon Web Services monitoring tools](#)

- [Aktivieren AWS Config](#)
- [Enable Amazon S3 server access logging](#)
- [Use CloudTrail](#)
- [Monitor Amazon Web Services security advisories](#)

Compliance-Validierung für Amazon S3

Die Sicherheit und Konformität von Amazon S3 wird von unabhängigen Prüfern im Rahmen mehrerer Prüfungen bewertet AWS Compliance-Programme, einschließlich der folgenden:

- System- und Organisationskontrollen (SOC)
- Datenschutzstandard der Zahlungskartenbranche (PCIDSS)
- Föderales Risiko- und Autorisierungsmanagementprogramm (FedRAMP)
- Gesetz über die Portabilität und Rechenschaftspflicht von Krankenversicherungen () HIPAA

AWS bietet eine häufig aktualisierte Liste von AWS Dienstleistungen im Rahmen spezifischer Compliance-Programme unter [AWS Dienstleistungen im Geltungsbereich des Compliance-Programms](#).

Prüfberichte von Drittanbietern stehen Ihnen zum Herunterladen zur Verfügung unter AWS Artifact. Weitere Informationen finden Sie unter Berichte [herunterladen in AWS Artifact](#).

Weitere Informationen zur AWS Compliance-Programme, siehe [AWS Compliance-Programme](#).

Welche Compliance-Verpflichtungen Sie bei der Verwendung von Amazon S3 haben, hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihrer Organisation und den geltenden Gesetzen und Vorschriften ab. Wenn Ihre Nutzung von Amazon S3 der Einhaltung von Standards wie HIPAA/PCI, oder Fed unterliegt/RAMP, AWS stellt Ressourcen zur Verfügung, um Ihnen zu helfen:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#), in denen architektonische Überlegungen und Schritte zur Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen erläutert werden AWS.
- [Architecting for HIPAA Security and Compliance](#) beschreibt, wie Unternehmen AWS um ihnen zu helfen, die HIPAA Anforderungen zu erfüllen.
- [AWS Die Compliance-Ressourcen](#) bieten verschiedene Arbeitsmappen und Leitfäden, die möglicherweise auf Ihre Branche und Ihren Standort zutreffen.
- [AWS Config](#) Mit können Sie bewerten, zu welchem Grad die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- [AWS Security Hub](#) bietet Ihnen einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb AWS und hilft Ihnen dabei, Ihre Einhaltung der Sicherheitsstandards und Best Practices der Branche zu überprüfen.

- [Verwenden der S3-Objektsperre](#) kann Ihnen helfen, die technischen Anforderungen der Aufsichtsbehörden für Finanzdienstleistungen zu erfüllen (wie z. B. das SECFINRA, und CFTC), wonach für bestimmte Arten von Buch- und Datensatzinformationen Datenspeicher mit einmaligem Schreiben und Lesen von vielen (WORM) Daten erforderlich sind.
- [Amazon S3 Inventory](#) können Sie bei der Prüfung und Meldung des Replikations- und Verschlüsselungsstatus Ihrer Objekte für Unternehmens-, Compliance- und regulatorische Anforderungen verwenden.

Ausfallsicherheit bei Amazon S3

Das Tool AWS Die globale Infrastruktur basiert auf Regionen und Availability Zones. AWS-Regionen bieten mehrere, physisch getrennte und isolierte Availability Zones, die mit geringer Latenz, hohem Durchsatz und hochredundanten Netzwerken miteinander verbunden sind. Diese Availability Zones bieten Ihnen eine effektive Methode zum Entwerfen und Betreiben von Anwendungen und Datenbanken. Availability Zones sind in noch größerem Ausmaß hochverfügbar und sie sind fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren. Wenn Sie Ihre Daten speziell über größere geografische Entfernungen replizieren müssen, können Sie diese Option verwenden, die das automatische [Überblick über das Replizieren von Objekten](#), asynchrone Kopieren von Objekten über Buckets in verschiedenen Buckets ermöglicht AWS-Regionen.

Jeder AWS-Region hat mehrere Availability Zones. Sie können Ihre Anwendungen über mehrere Availability Zones in derselben Region bereitstellen, um eine bessere Fehlertoleranz und niedriger Latenz zu erzielen. Availability Zones sind mit schnellen, privaten Glasfasernetzwerken verbunden. Dies ermöglicht Ihnen die Nutzung von Anwendungen, für die ein automatischer, unterbrechungsfreier Failover zwischen den Availability Zones eingerichtet ist.

Weitere Informationen zur AWS-Regionen und Availability Zones, siehe [AWS Globale Infrastruktur](#).

Zusätzlich zu AWS Amazon S3 ist eine globale Infrastruktur und bietet mehrere Funktionen, die Sie bei Ihren Anforderungen an Datenstabilität und Datensicherung unterstützen.

Lebenszyklus-Konfiguration

Eine Lebenszyklus-Konfiguration besteht aus einer Reihe von Regeln, mit denen Aktionen definiert werden, die Amazon S3 auf eine Gruppe von Objekten anwendet. Mithilfe der Konfigurationsregeln für den Lebenszyklus können Sie Amazon S3 anweisen, Objekte in kostengünstigere Speicherklassen zu übergeben bzw. zu archivieren oder zu löschen. Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Versioning

Das Versioning ermöglicht Ihnen, mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Sie können Versioning verwenden, um sämtliche Versionen aller Objekte in Ihrem Amazon S3 Bucket zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leicht wiederherstellen. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

S3-Objektsperre

Sie können S3 Object Lock verwenden, um Objekte mit einem Write-Once-Read-Many (WORM) -Modell zu speichern. Mit der S3-Objektsperre können Sie für einen festen Zeitraum oder auf unbegrenzte Zeit verhindern, dass ein Objekt gelöscht oder überschrieben wird. Mit S3 Object Lock können Sie gesetzliche Anforderungen erfüllen, die WORM Speicherplatz erfordern, oder einfach eine zusätzliche Schutzebene vor Objektänderungen und -lösungen hinzufügen. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Speicherklassen

Amazon S3 bietet je nach den Anforderungen Ihrer Workload eine Reihe von Speicherklassen an. Die Speicherklassen S3 Standard-IA und S3 One Zone-IA sind für Daten konzipiert, auf die Sie etwa einmal im Monat zugreifen und auf Millisekunden zugreifen müssen. Die Speicherklasse S3 Glacier Instant Retrieval ist für langlebige Archivdaten konzipiert, auf die Sie mit Millisekunden-Zugriff zugreifen, auf den Sie etwa einmal pro Quartal zugreifen. Für Archivdaten, die keinen sofortigen Zugriff erfordern, wie zum Beispiel Backups, können Sie die Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive verwenden. Weitere Informationen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Die folgenden bewährten Sicherheitsmethoden umfassen ebenfalls die Ausfallsicherheit:

- [Enable versioning](#)
- [Consider Amazon S3 cross-region replication](#)
- [Identify and audit all your Amazon S3 buckets](#)

Verschlüsselung von Amazon-S3-Sicherungen

Wenn Sie Backups mit Amazon S3 speichern, hängt die Verschlüsselung Ihrer Backups von der Konfiguration dieser Buckets ab. Die Amazon-S3-Standard-Verschlüsselung bietet eine Methode zum Festlegen des Verhaltens der Standard-Verschlüsselung für einen S3-Bucket. Sie können die Standard-Verschlüsselung in einem Bucket festlegen, sodass alle Objekte beim Speichern im Bucket verschlüsselt werden. Die Standardverschlüsselung unterstützt Schlüssel, die gespeichert sind in AWS KMS (SSE-KMS). Weitere Informationen finden Sie unter [Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets](#).

Weitere Informationen zu Versioning und Objektsperre finden Sie in den folgenden Themen: [Verwenden der Versioning in S3-Buckets](#) [Verwenden der S3-Objektsperre](#)

Infrastruktursicherheit in Amazon S3

Als verwalteter Service ist Amazon S3 geschützt durch AWS Verfahren zur globalen Netzwerksicherheit, die in der Sicherheitssäule der beschrieben sind [AWS Well-Architected Framework](#).

Der Zugriff auf Amazon S3 über das Netzwerk erfolgt über AWS veröffentlicht APIs. Clients müssen Transport Layer Security (TLS) 1.2 unterstützen. Wir empfehlen, auch TLS 1.3 zu unterstützen. (Weitere Informationen zu dieser Empfehlung finden Sie unter [Schneller AWS Cloud-Verbindungen mit TLS 1.3](#) auf dem AWS Sicherheits-Blog.) Kunden müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral Diffie-Hellman (E) oder Elliptic Curve Diffie-Hellman Ephemeral (DHE) unterstützen. ECDHE Darüber hinaus müssen Anfragen wie folgt signiert werden AWS Signatur V4 oder AWS Signatur V2, für die gültige Anmeldeinformationen angegeben werden müssen.

Diese APIs können von jedem Netzwerkstandort aus aufgerufen werden. Da Amazon S3 jedoch ressourcenbasierte Zugriffsrichtlinien unterstützt, kann es zu Einschränkungen bezüglich der Quell-IP-Adresse kommen. Sie können auch Amazon S3 S3-Bucket-Richtlinien verwenden, um den Zugriff auf Buckets von bestimmten Virtual Private Cloud (VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dadurch wird der Netzwerkzugriff auf einen bestimmten Amazon S3 S3-Bucket effektiv nur von den spezifischen VPC innerhalb des AWS Netzwerk. Weitere Informationen finden Sie unter [Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien](#).

Die folgenden bewährten Sicherheitsmethoden beziehen sich ebenfalls auf die Sicherheit der Infrastruktur von Amazon S3:

- [Consider VPC endpoints for Amazon S3 access](#)
- [Identify and audit all your Amazon S3 buckets](#)

Konfigurations- und Schwachstellenanalyse in Amazon S3

AWS kümmert sich um grundlegende Sicherheitsaufgaben wie das Patchen von Gastbetriebssystemen (OS) und Datenbanken, die Firewall-Konfiguration und die Notfallwiederherstellung. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert. Weitere Informationen finden Sie in den folgenden Ressourcen:

- [Compliance-Validierung für Amazon S3](#)
- [Modell der übergreifenden Verantwortlichkeit](#)
- [Amazon Web Services – Übersicht über Sicherheitsverfahren](#)

Die folgenden bewährten Sicherheitsmethoden beinhalten auch die Adresskonfiguration und Schwachstellenanalyse in Amazon S3:

- [Identify and audit all your Amazon S3 buckets](#)
- [Aktiviert AWS Config](#)

Bewährte Methoden für die Sicherheit in Amazon S3

Amazon S3 enthält eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden stellen allgemeine Richtlinien und keine vollständige Sicherheitslösung dar. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Empfehlungen und nicht als bindend ansehen.

Themen

- [Bewährte Methoden für die Sicherheit in Amazon S3](#)
- [Bewährte Methoden zur Überwachung und Prüfung von Amazon S3](#)

Bewährte Methoden für die Sicherheit in Amazon S3

Die folgenden bewährten Methoden für Amazon S3 können dazu beitragen, Sicherheitsvorfälle zu verhindern.

Deaktivieren Sie die Zugriffskontrolllisten (ACLs)

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren können. ACLs Standardmäßig ist für Object Ownership die Einstellung Bucket Owner enforced festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

In den meisten modernen Anwendungsfällen in Amazon S3 ist die Verwendung von [Zugriffskontrolllisten \(ACLs\)](#) nicht mehr erforderlich. Wir empfehlen die Deaktivierung ACLs, außer in ungewöhnlichen Fällen, in denen Sie den Zugriff für jedes Objekt einzeln kontrollieren müssen. Um jedes Objekt in Ihrem Bucket zu deaktivieren ACLs und die Eigentümerschaft dafür zu übernehmen, wenden Sie für S3 Object Ownership die Einstellung Bucket Ownerforced an. Wenn Sie die Option deaktivieren ACLs, können Sie ganz einfach einen Bucket mit Objekten verwalten, die von verschiedenen Anbietern hochgeladen wurden AWS-Konten.

Wann deaktiviert ACLs sind, basiert die Zugriffskontrolle für Ihre Daten auf Richtlinien wie den folgenden:

- AWS Identity and Access Management (IAM) Benutzerrichtlinien

- S3-Bucket-Richtlinien
- Endpunktrichtlinien für virtuelle private Clouds (VPC)
- AWS Organizations Richtlinien zur Dienststeuerung (SCPs)

Die Deaktivierung ACLs vereinfacht die Verwaltung und Prüfung von Berechtigungen. ACLs sind für neue Buckets standardmäßig deaktiviert. Sie können sie auch ACLs für bestehende Buckets deaktivieren. Wenn Sie über einen vorhandenen Bucket verfügen, der bereits Objekte enthält, ACLs sind das Objekt und der Bucket nach der Deaktivierung ACLs nicht mehr Teil des Zugriffsevaluierungsprozesses. Stattdessen wird der Zugriff auf der Grundlage von Richtlinien gewährt oder verweigert.

Stellen Sie vor der Deaktivierung sicher ACLs, dass Sie Folgendes tun:

- Überprüfen Sie Ihre Bucket-Richtlinie, um sicherzustellen, dass sie alle Möglichkeiten abdeckt, wie Sie außerhalb Ihres Kontos Zugriff auf Ihren Bucket gewähren möchten.
- Setzen Sie Ihren Bucket ACL auf die Standardeinstellung zurück (volle Kontrolle hat der Bucket-Besitzer).

Nach der Deaktivierung ACLs treten die folgenden Verhaltensweisen auf:

- Ihr Bucket akzeptiert nur PUT Anfragen, die keine PUT Oder-Anfragen angeben, für die der Bucket-Besitzer die volle Kontrolle hat ACLs. ACL Dazu ACLs gehören `bucket-owner-full-control` vorgefertigte ACL oder gleichwertige Formen ACL, die in ausgedrückt werden XML.
- Bestehende Anwendungen, die die volle Kontrolle über ACLs den Bucket Owner unterstützen, haben keine Auswirkungen.
- PUT Anfragen, die andere beinhalten ACLs (z. B. benutzerdefinierte Zuschüsse für bestimmte AWS-Konten) schlagen fehl und geben einen HTTP Statuscode 400 (Bad Request) mit dem Fehlercode zurück `AccessControlListNotSupported`.

Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Achten Sie darauf, dass für Ihre Amazon-S3-Buckets die korrekten Richtlinien gelten und dass sie nicht öffentlich zugänglich sind.

Sofern es nicht unbedingt notwendig ist, dass jemand Ihren S3-Bucket im Internet lesen oder darin schreiben kann, stellen Sie sicher, dass der S3-Bucket nicht öffentlich ist. Im Folgenden werden einige Maßnahmen zum Blockieren des öffentlichen Zugriffs aufgeführt:

- Verwenden Sie S3 Block Public Access. Mit S3 Block Public Access können Sie problemlos zentrale Kontrollen zur Beschränkung des öffentlichen Zugriffs auf Ihre Amazon-S3-Ressourcen einrichten. Diese zentralen Kontrollen werden unabhängig davon durchgesetzt, wie die Ressourcen erstellt werden. Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).
- Identifizieren Sie Amazon-S3-Bucket-Richtlinien, die eine Platzhalteridentität zulassen, z. B. "Principal": "*" (was praktisch „jeder“ bedeutet). Halten Sie auch nach Richtlinien Ausschau, die die Platzhalteraktion "*" zulassen (was einem Benutzer erlaubt, jede Aktion im Amazon-S3-Bucket durchzuführen).
- Suchen Sie in ähnlicher Weise nach Amazon S3 S3-Bucket-Zugriffskontrolllisten (ACLs), die Lese-, Schreib- oder Vollzugriff für „Jeder“ oder „Jeder authentifizierte Benutzer“ gewähren AWS Benutzer.“
- Verwenden Sie den ListBuckets API Vorgang, um alle Ihre Amazon S3 S3-Buckets zu scannen. Ermitteln Sie dann mit GetBucketAcl, GetBucketWebsite und GetBucketPolicy, ob der Bucket über richtlinienkonforme Zugriffskontrollen und Konfiguration verfügt.
- Verwenden von [AWS Trusted Advisor](#) um Ihre Amazon S3 S3-Implementierung zu überprüfen.
- Erwägen Sie die Implementierung laufender detektiver Kontrollen mithilfe der [s3-bucket-public-read-prohibited](#) und [s3-bucket-public-write-prohibited](#) verwaltet AWS-Config-Regeln.

Weitere Informationen finden Sie unter [Identity and Access Management für Amazon S3](#).

Identifizieren Sie mithilfe von Amazon potenzielle Bedrohungen für Ihre Amazon S3-Buckets GuardDuty

[Amazon GuardDuty](#) ist ein Service zur Bedrohungserkennung, der potenzielle Bedrohungen für Ihre Konten, Container, Workloads und die Daten in Ihrer AWS Umgebung. Mithilfe von Modellen für maschinelles Lernen (ML) und Funktionen zur Erkennung von Anomalien und Bedrohungen überwacht Amazon GuardDuty kontinuierlich verschiedene Datenquellen, um potenzielle Sicherheitsrisiken und böswillige Aktivitäten in Ihrer Umgebung zu identifizieren und zu priorisieren. Wenn Sie es aktivieren GuardDuty, bietet es Bedrohungserkennung für grundlegende Datenquellen, darunter [AWS CloudTrail Verwaltungsereignisse](#), VPC Ablaufprotokolle und DNS Protokolle. Um die Bedrohungserkennung auf Ereignisse auf Datenebene in S3-Buckets auszudehnen, können Sie die [GuardDuty S3-Schutzfunktion](#) aktivieren. Diese Funktion erkennt Bedrohungen wie Datenexfiltration und verdächtigen Zugriff auf S3-Buckets über Tor-Knoten. GuardDuty erstellt auch ein normales Basismuster in Ihrer Umgebung. Wenn es ein potenziell anomales Verhalten identifiziert, stellt es Kontextinformationen bereit, die Ihnen helfen, den

potenziell gefährdeten S3-Bucket zu beheben, oder AWS Anmeldeinformationen. Weitere Informationen finden Sie unter [GuardDuty](#).

Implementieren des Zugriffs mit geringsten Berechtigungen

Bei der Vergabe von Berechtigungen entscheiden Sie, wer welche Berechtigungen für welche Amazon-S3-Ressourcen erhält. Sie aktivieren die spezifischen Aktionen, die daraufhin für die betreffenden Ressourcen erlaubt sein sollen. Aus diesem Grund sollten Sie nur Berechtigungen gewähren, die zum Ausführen einer Aufgabe erforderlich sind. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung zum Reduzieren des Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Die folgenden Tools stehen zur Implementierung der geringstmöglichen Zugriffsrechte zur Verfügung:

- [Politische Maßnahmen für Amazon S3](#) und [Berechtigungsgrenzen für IAM Entitäten](#)
- [So funktioniert Amazon S3 mit IAM](#)
- [Übersicht über die Zugriffskontrollliste \(ACL\)](#)
- [Service-Kontrollrichtlinien](#)

Hilfreiche Informationen zur Entscheidung, welche der vorherigen Mechanismen Sie auswählen sollten, finden Sie unter [Identity and Access Management für Amazon S3](#).

Verwenden Sie IAM Rollen für Anwendungen und AWS-Services für die Amazon S3 S3-Zugriff erforderlich ist

Für Anwendungen, die auf Amazon EC2 oder anderen laufen AWS-Services um auf Amazon S3 S3-Ressourcen zugreifen zu können, müssen sie gültige AWS Anmeldeinformationen in ihren AWS API-Anfragen. Wir empfehlen, nicht zu speichern AWS Anmeldeinformationen direkt in der Anwendung oder EC2 Amazon-Instance. Dabei handelt es sich um langfristige Anmeldeinformationen, die nicht automatisch rotiert werden und bedeutende geschäftliche Auswirkungen haben könnten, wenn sie kompromittiert werden.

Verwenden Sie stattdessen eine IAM Rolle, um temporäre Anmeldeinformationen für Anwendungen oder Dienste zu verwalten, die auf Amazon S3 zugreifen müssen. Wenn Sie eine Rolle verwenden, müssen Sie keine langfristigen Anmeldeinformationen (wie einen Benutzernamen und ein Passwort oder Zugangsschlüssel) an eine EC2 Amazon-Instance verteilen oder AWS-Service, wie beispielsweise AWS Lambda. Die Rolle stellt temporäre Berechtigungen bereit, die Anwendungen verwenden können, wenn sie andere Benutzer aufrufen AWS Ressourcen schätzen.

Weitere Informationen finden Sie in den folgenden Themen im IAMBenutzerhandbuch:

- [IAMRollen](#)
- [Gängige Szenarien für Rollen: Benutzer, Anwendungen und Services](#)

Erwägen der Verschlüsselung von Data-at-Rest

Es gibt die folgenden Optionen, Daten im Ruhezustand in Amazon S3 zu schützen:

- Serverseitige Verschlüsselung — Für alle Amazon S3 S3-Buckets ist standardmäßig eine Verschlüsselung konfiguriert, und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Amazon S3 bietet auch diese serverseitigen Verschlüsselungsoptionen:

- Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS)
- Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (-) DSSE KMS
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Weitere Informationen finden Sie unter [Schützen von Daten mit serverseitiger Verschlüsselung](#).

- Clientseitige Verschlüsselung – Sie können Daten clientseitig verschlüsseln und die verschlüsselten Daten auf Amazon S3 hochladen. In diesem Fall verwalten Sie den Verschlüsselungsprozess, die Verschlüsselungsschlüssel und die zugehörigen Tools. Genau wie die serverseitige Verschlüsselung kann die client-seitige Verschlüsselung das Risiko reduzieren, indem die Daten mit einem Schlüssel verschlüsselt werden, der durch einen anderen Mechanismus gespeichert wird als der, der die Daten selbst speichert

Amazon S3 bietet mehrere clientseitige Verschlüsselungsoptionen. Weitere Informationen finden Sie unter [Schützen von Daten mithilfe der clientseitigen Verschlüsselung](#).

Erzwingen der Verschlüsselung von Daten während der Übertragung

Sie können HTTPS (TLS) verwenden, um zu verhindern, dass potenzielle Angreifer den Netzwerkverkehr mit oder ähnlichen Angriffen abhören oder manipulieren. person-in-the-middle

Wir empfehlen, nur verschlüsselte Verbindungen über HTTPS () TLS zuzulassen, indem Sie [aws:SecureTransport](#) Bedingung in Ihren Amazon S3 S3-Bucket-Richtlinien.

⚠ Important

Wir empfehlen, dass Ihre Anwendung Amazon TLS S3-Zertifikate nicht als AWS unterstützt das Anheften von öffentlich vertrauenswürdigen Zertifikaten nicht. S3 erneuert Zertifikate automatisch und die Verlängerung kann jederzeit vor Ablauf des Zertifikats erfolgen. Durch die Erneuerung eines Zertifikats wird ein neues öffentlich-privates key pair generiert. Wenn Sie ein S3-Zertifikat angeheftet haben, das kürzlich mit einem neuen öffentlichen Schlüssel erneuert wurde, können Sie erst dann eine Verbindung zu S3 herstellen, wenn Ihre Anwendung das neue Zertifikat verwendet.

Erwägen Sie auch die Implementierung fortlaufender Erkennungskontrollen mithilfe der [s3-bucket-ssl-requests-only](#) verwaltet AWS Config Regel.

Erwägen der S3-Objektsperre

Mit S3 Object Lock können Sie Objekte mithilfe eines „Write Once Read Many“ (WORM) -Modells speichern. Die S3-Objektsperre kann unbeabsichtigtes oder unsachgemäßes Löschen von Daten verhindern. Beispielsweise können Sie S3 Object Lock verwenden, um Ihre AWS CloudTrail Logs.

Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Aktivieren der S3-Versionsverwaltung

Die S3-Versionsverwaltung ermöglicht Ihnen, mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Sie können Versioning verwenden, um sämtliche Versionen aller Objekte in Ihrem Bucket zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leicht wiederherstellen.


Erwägen Sie auch die Implementierung fortlaufender Detektivkontrollen mithilfe der [s3-bucket-versioning-enabled](#) verwaltet AWS Config Regel.

Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Erwägen einer regionsübergreifenden S3-Replikation

Auch wenn Amazon S3 Ihre Daten standardmäßig in mehreren entfernten Availability Zones speichert, machen es die Compliance-Anforderungen möglicherweise erforderlich, Daten in

noch größeren Entfernungen zu speichern. Mit der regionsübergreifenden S3-Replikation (CRR) können Sie Daten zwischen entfernten Standorten replizieren AWS-Regionen um dabei zu helfen, diese Anforderungen zu erfüllen. CRR ermöglicht das automatische, asynchrone Kopieren von Objekten in verschiedenen Buckets AWS-Regionen. Weitere Informationen finden Sie unter [Überblick über das Replizieren von Objekten](#).

 Note

CRR erfordert, dass sowohl für den S3-Quell- als auch für den Ziel-S3-Bucket die Versionierung aktiviert ist.

Erwägen Sie auch die Implementierung fortlaufender Erkennungskontrollen mithilfe von [s3-bucket-replication-enabled](#) verwaltet AWS Config Regel.

Erwägen Sie die Verwendung von VPC Endpunkten für den Zugriff auf Amazon S3

Ein Virtual Private Cloud (VPC) -Endpunkt für Amazon S3 ist eine logische Einheit innerhalb einer VPC, die nur Konnektivität zu Amazon S3 ermöglicht. VPC Endpunkte können dazu beitragen, zu verhindern, dass Datenverkehr das offene Internet durchquert.

VPC Endpunkte für Amazon S3 bieten mehrere Möglichkeiten, den Zugriff auf Ihre Amazon S3 S3-Daten zu kontrollieren:

- Mithilfe von S3-Bucket-Richtlinien können Sie die Anfragen, Benutzer oder Gruppen steuern, die über einen bestimmten VPC Endpunkt zugelassen werden.
- Mithilfe von S3-Bucket-Richtlinien können Sie steuern, welche VPCs oder welche VPC Endpunkte Zugriff auf Ihre S3-Buckets haben.
- Sie können dazu beitragen, Datenexfiltration zu verhindern, indem Sie ein Gerät verwenden VPC, das kein Internet-Gateway hat.

Weitere Informationen finden Sie unter [Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien](#).

Verwaltet verwenden AWS Sicherheitsdienste zur Überwachung der Datensicherheit

Mehrere verwaltet AWS Sicherheitsservices können Ihnen helfen, Sicherheits- und Compliance-Risiken für Ihre Amazon S3 S3-Daten zu identifizieren, zu bewerten und zu überwachen. Diese Services können Ihnen auch dabei helfen, Ihre Daten vor diesen Risiken zu schützen. Zu diesen Services gehören automatisierte Erkennungs-, Überwachungs- und Schutzfunktionen, die so

konzipiert sind, dass sie von Amazon S3 S3-Ressourcen auf eine einzige Lösung skaliert werden können AWS-Konto bis hin zu Ressourcen für Organisationen mit Tausenden von Konten.

Weitere Informationen finden Sie unter [Überwachung der Datensicherheit mit Managed AWS Sicherheitsdienste](#).

Bewährte Methoden zur Überwachung und Prüfung von Amazon S3

Die folgenden bewährten Methoden für Amazon S3 können dabei helfen, potenzielle Sicherheitsschwächen und Vorfälle zu erkennen.

Identifizieren und Prüfen aller Ihrer Amazon-S3-Buckets

Die Identifikation Ihrer IT-Assets ist ein wichtiger Aspekt von Governance und Sicherheit. Es ist erforderlich, dass Sie alle Ihre Amazon-S3-Ressourcen überblicken, um ihren Sicherheitsstatus beurteilen und Maßnahmen gegen potenzielle Schwachstellen ergreifen zu können. Zum Prüfen Ihrer Ressourcen empfehlen wir Folgendes:

- Verwenden Sie den Tag-Editor, um sicherheits- und prüfungsrelevante Ressourcen zu identifizieren und zu markieren. Nutzen Sie dann diese Markierungen für die Suche nach den entsprechenden Ressourcen. Weitere Informationen finden Sie unter [Suchen nach Ressourcen, die im Tagging markiert](#) werden sollen AWS Benutzerhandbuch für Ressourcen.
- Verwenden Sie S3 Inventory für die Prüfung und Meldung des Replikations- und Verschlüsselungsstatus Ihrer Objekte für Unternehmens-, Compliance- und regulatorische Anforderungen. Weitere Informationen finden Sie unter [Amazon S3 Inventory](#).
- Erstellen Sie Ressourcengruppen für Ihre Amazon-S3-Ressourcen. Weitere Informationen finden Sie unter [Was sind Ressourcengruppen?](#) in der AWS Resource Groups Benutzerleitfaden.

Implementieren Sie die Überwachung mithilfe von AWS Überwachungstools

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Sicherheit, Verfügbarkeit und Leistung von Amazon S3 und Ihrer AWS Lösungen. AWS bietet verschiedene Tools und Dienste, mit denen Sie Amazon S3 und Ihre anderen überwachen können AWS-Services. Sie können beispielsweise CloudWatch Amazon-Metriken für Amazon S3 überwachen, insbesondere die DeleteRequests Kennzahlen PutRequests GetRequests4xxErrors,, und. Weitere Informationen erhalten Sie unter [Metriken mit Amazon überwachen CloudWatch](#) und [Überwachen von Amazon S3](#).

Ein zweites Beispiel finden Sie unter [Beispiel: Amazon-S3-Bucket-Aktivität](#). In diesem Beispiel wird beschrieben, wie ein CloudWatch Alarm erstellt wird, der ausgelöst wird, wenn ein Amazon S3 API S3-Aufruf an PUT DELETE eine Bucket-Richtlinie, einen Bucket-Lebenszyklus oder eine Bucket-Replikationskonfiguration oder an PUT einen Bucket erfolgtACL.

Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an einen Bucket gestellt wurden. Server-Zugriffsprotokolle können Sie bei Sicherheits- und Zugriffsprüfungen unterstützen; sie geben Informationen über Ihren Kundenstamm und helfen beim Verständnis der Amazon-S3-Rechnung. Weitere Informationen zur Aktivierung der Server-Zugriffsprotokollierung finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

Erwägen Sie auch die Implementierung fortlaufender detektiver Kontrollen mithilfe von [s3-bucket-logging-enabled](#) AWS Config verwaltete Regel.

Benutzen AWS CloudTrail

AWS CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service auf Amazon S3. Sie können die von gesammelten Informationen verwenden CloudTrail , um Folgendes zu ermitteln:

- Die Anforderung, die an Amazon S3 gestellt wurde
- Die IP-Adresse, von der die Anforderung erfolgt ist
- Wer die Anforderung gestellt hat
- Wann die Anforderung gestellt wurde
- Zusätzliche Details zur Anforderung

Sie können beispielsweise CloudTrail Einträge für PUT Aktionen identifizieren, die sich auf den Datenzugriff auswirkenPutBucketAc1, insbesondere PutObjectAc1PutBucketPolicy, undPutBucketWebsite.

Wenn Sie Ihre einrichten AWS-Konto, CloudTrail ist standardmäßig aktiviert. Sie können aktuelle Ereignisse in der CloudTrail Konsole anzeigen. Um eine fortlaufende Aufzeichnung der Aktivitäten und Ereignisse für Ihre Amazon S3 S3-Buckets zu erstellen, können Sie in der CloudTrail Konsole einen Trail erstellen. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen](#) im AWS CloudTrail Benutzerleitfaden.

Wenn Sie einen Trail erstellen, können Sie ihn so konfigurieren, CloudTrail dass Datenereignisse protokolliert werden. Datenereignisse sind Datensätze von Ressourcen-Vorgänge, die auf oder innerhalb einer Ressource ausgeführt werden. In Amazon S3 zeichnen Datenereignisse API Aktivitäten auf Objektebene für einzelne Buckets auf. CloudTrail unterstützt eine Teilmenge von Amazon S3 API S3-Operationen auf Objektebene, wie `GetObject`, und `DeleteObject`. `PutObject` Weitere Informationen zur CloudTrail Funktionsweise mit Amazon S3 finden Sie unter [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#). In der Amazon-S3-Konsole können Sie Ihre S3-Buckets auch dafür konfigurieren, [Aktivieren der CloudTrail Ereignisprotokollierung für S3-Buckets und -Objekte](#).

AWS Config stellt eine verwaltete Regel (`cloudtrail-s3-dataevents-enabled`) bereit, mit der Sie bestätigen können, dass mindestens ein CloudTrail Trail Datenereignisse für Ihre S3-Buckets protokolliert. Weitere Informationen finden Sie unter [cloudtrail-s3-dataevents-enabled](#) in der AWS Config Leitfaden für Entwickler.

Aktivieren AWS Config

In einigen der in diesem Thema aufgeführten bewährten Methoden wird empfohlen, Folgendes zu erstellen AWS Config Regeln. AWS Config hilft Ihnen bei der Bewertung, Prüfung und Bewertung der Konfigurationen Ihrer AWS Ressourcen schätzen. AWS Config überwacht die Ressourcenkonfigurationen, sodass Sie die aufgezeichneten Konfigurationen mit den gewünschten sicheren Konfigurationen vergleichen können. Mit AWS Config, Sie können Folgendes tun:

- Überprüfen Sie Änderungen an Konfigurationen und Beziehungen zwischen AWS Ressourcen
- Untersuchen detaillierter Ressourcenkonfigurationsverläufe
- Bestimmen der allgemeinen Compliance im Hinblick auf die Konfigurationen, die in Ihren internen Leitlinien angegeben sind

Die Verwendung von AWS Config kann Ihnen dabei helfen, Compliance-Prüfungen, Sicherheitsanalysen, das Änderungsmanagement und die betriebliche Fehlerbehebung zu vereinfachen. Weitere Informationen finden Sie unter [Einrichtung AWS Config mit der Konsole](#) im AWS Config Leitfaden für Entwickler. Wenn Sie die aufzuzeichnenden Ressourcentypen angeben, stellen Sie sicher, dass Sie Amazon-S3-Ressourcen mit einbeziehen.

Important

AWS Config managed rules unterstützt bei der Evaluierung von Amazon S3 S3-Ressourcen nur Buckets für allgemeine Zwecke. AWS Config zeichnet keine

Konfigurationsänderungen für Verzeichnis-Buckets auf. Weitere Informationen finden Sie unter [AWS Config Verwaltete Regeln](#) und [Liste von AWS Config Verwaltete Regeln](#) in der AWS Config Leitfaden für Entwickler.

Für ein Beispiel für die Verwendung AWS Config, siehe [Wie benutzt man AWS Config um Amazon S3 S3-Buckets zu überwachen und darauf zu reagieren, die öffentlichen Zugriff](#) auf den AWS Blog zum Thema Sicherheit.

Entdecken von sensiblen Daten mithilfe von Amazon Macie

Amazon Macie ist ein Sicherheitsservice, der sensible Daten mithilfe von Machine Learning und Musterabgleich erkennt. Macie bietet Einblick in Datensicherheitsrisiken und ermöglicht automatischen Schutz vor diesen Risiken. Mit Macie können Sie die Erkennung und Meldung sensibler Daten in Ihrem Amazon-S3-Datenbestand automatisieren, um die Daten besser zu verstehen, die Ihre Organisation in S3 speichert.

Um sensible Daten mit Macie zu erkennen, können Sie integrierte Kriterien und Techniken verwenden, die darauf ausgelegt sind, eine lange und wachsende Liste vertraulicher Datentypen für viele Länder und Regionen zu erkennen. Zu diesen sensiblen Datentypen gehören mehrere Arten von personenbezogenen Daten (PII), Finanzdaten und Anmeldeinformationen. Sie können auch benutzerdefinierte Kriterien verwenden, die Sie definieren – reguläre Ausdrücke, die Textmuster zum Abgleichen definieren, und optional Zeichenfolgen und Näherungsregeln, um die Ergebnisse zu verfeinern.

Wenn Macie sensible Daten in einem S3-Objekt entdeckt, generiert es eine Sicherheitserkenntnis, um Sie zu benachrichtigen. Diese Erkenntnis enthält Informationen über das betroffene Objekt, die Art und Häufigkeit der Vorkommen der sensiblen Daten, die Macie gefunden hat, sowie zusätzliche Details, die Ihnen bei der Untersuchung des betroffenen S3-Buckets und -Objekts helfen. Weitere Informationen finden Sie im [Benutzerhandbuch für Amazon Macie](#).

Verwenden von S3 Storage Lens

S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

Mit S3 Storage Lens können Sie Metriken verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben

oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz und die Zugriffsverwaltung implementieren und die Leistung von Anwendungs-Workloads verbessern.

Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, abgebrochen werden. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. Weitere Informationen finden Sie unter [Grundlegendes zu Amazon S3 Storage Lens](#).

Überwachen AWS Sicherheitshinweise

Wir empfehlen Ihnen, regelmäßig die Sicherheitshinweise zu lesen, die unter veröffentlicht werden Trusted Advisor für Ihre AWS-Konto. Achten Sie insbesondere auf Warnungen zu Amazon S3 S3-Buckets mit „Open-Access-Berechtigungen“. Sie können dies programmgesteuert tun, indem Sie [describe-trusted-advisor-checks](#).

Überwachen Sie außerdem aktiv die primäre E-Mail-Adresse, die für jede Ihrer E-Mails registriert ist AWS-Konten. AWS verwendet diese E-Mail-Adresse, um Sie über neu auftretende Sicherheitsprobleme zu informieren, die Sie betreffen könnten.

AWS betriebliche Probleme mit weitreichenden Auswirkungen werden auf der [AWS Health Dashboard - Gesundheit im Service](#). Betriebliche Probleme werden auch auf einzelne Konten verbucht, und zwar über AWS Health Dashboard Weitere Informationen hierzu finden Sie unter [.AWS Health Dokumentation](#).

Überwachung der Datensicherheit mit Managed AWS Sicherheitsdienste

Mehrere verwaltete AWS Sicherheitservices können Ihnen helfen, Sicherheits- und Compliance-Risiken für Ihre Amazon S3 S3-Daten zu identifizieren, zu bewerten und zu überwachen. Sie können Sie auch dabei unterstützen, Ihre Daten vor diesen Risiken zu schützen. Zu diesen Services gehören automatisierte Erkennungs-, Überwachungs- und Schutzfunktionen, die so konzipiert sind, dass sie von Amazon S3 S3-Ressourcen auf eine einzige Lösung skaliert werden können AWS-Konto bis hin zu Ressourcen für Organisationen mit Tausenden von AWS-Konten.

AWS Erkennungs- und Reaktionsdienste können Ihnen dabei helfen, potenzielle Fehlkonfigurationen, Bedrohungen oder unerwartetes Verhalten im Sicherheitsbereich zu identifizieren, sodass Sie schnell auf potenziell unautorisierte oder bösartige Aktivitäten in Ihrer Umgebung reagieren können. AWS Datenschutzdienste können Ihnen helfen, Ihre Daten, Konten und Workloads zu überwachen und vor unbefugtem Zugriff zu schützen. Sie können Ihnen auch dabei helfen, sensible Daten, wie z. B. persönlich identifizierbare Informationen (PII), in Ihrem Amazon S3 S3-Datenbestand zu entdecken.

Um Ihnen bei der Identifizierung und Bewertung von Datensicherheits- und Compliance-Risiken zu helfen, verwaltet AWS Sicherheitsdienste generieren Ergebnisse, um Sie über potenzielle Sicherheitsereignisse oder Probleme mit Ihren Amazon S3 S3-Daten zu informieren. Die Erkenntnisse enthalten relevante Informationen, anhand derer Sie diese Risiken untersuchen, bewerten und entsprechend Ihren Vorfallreaktions-Workflows und -richtlinien handeln können. Über die einzelnen Services können Sie direkt auf die Erkenntnisdaten zugreifen. Sie können die Daten auch an andere Anwendungen, Dienste und Systeme senden, z. B. an Ihr System zur Verwaltung von Sicherheitsvorfällen und Ereignissen (SIEM).

Um die Sicherheit Ihrer Amazon S3 S3-Daten zu überwachen, sollten Sie die Verwendung dieser verwalteten Daten in Betracht ziehen AWS Sicherheitsdienste.

Amazon GuardDuty

Amazon GuardDuty ist ein Dienst zur Bedrohungserkennung, der Ihre AWS-Konten sowie Workloads für böswillige Aktivitäten und liefert detaillierte Sicherheitsinformationen für Transparenz und Problembeseitigung.

Amazon GuardDuty bietet zwei Schutzpläne an, mit denen Sie potenzielle Bedrohungen in Ihren S3-Buckets erkennen können.

GuardDuty S3-Schutz

- Wenn Sie S3-Schutz aktivieren, werden GuardDuty Monitore AWS CloudTrail Verwaltung und Datenereignisse für Ihre Amazon S3 S3-Ressourcen. GuardDuty analysiert dann diese Ereignisse wie mögliche Datenexfiltration und Datenvernichtung. GuardDuty Nutzt Threat-Intelligence-Feeds und maschinelles Lernen, um fundierte Analysen durchzuführen und potenzielle Sicherheitsrisiken zu identifizieren.
- GuardDuty kann verschiedene Arten von Aktivitäten für Ihre Amazon S3 S3-Ressourcen überwachen. Beispielsweise umfassen CloudTrail Verwaltungsereignisse für Amazon S3 Operationen auf Bucket-Ebene, wie `ListBucketsDeleteBucket`, und `PutBucketReplication`; CloudTrail Datenereignisse für Amazon S3 beinhalten Operationen auf Objektebene wie `GetObject`, und `ListObjectsPutObject` Wenn ungewöhnliche oder potenziell bösartige Aktivitäten GuardDuty erkannt werden, wird ein Ergebnis generiert, das Sie benachrichtigt.
- Weitere Informationen finden Sie unter [GuardDuty S3 Protection](#) im GuardDuty Amazon-Benutzerhandbuch.

Malware-Schutz für S3

- Aktivieren Sie diesen Schutzplan, um neu hochgeladene Objekte in Ihre ausgewählten S3-Buckets auf potenzielle Malware zu scannen. Sie können den Umfang des Malware-Scans auch auf bestimmte Objektpräfixe beschränken.
- Diese Funktion bietet Funktionen wie die automatische Überwachung des Scanstatus mit Amazon EventBridge, die Bewertung von Scan-Metriken mit Amazon CloudWatch und die optionale Kennzeichnung bösartiger S3-Objekte.
- GuardDuty bietet die Option, den Malware-Schutz für S3 zu aktivieren, ohne den GuardDuty Service aktivieren zu müssen. Weitere Informationen finden Sie unter [GuardDuty Malware-Schutz für S3](#) im GuardDuty Amazon-Benutzerhandbuch.

Amazon Detective

Amazon Detective vereinfacht den Ermittlungsprozess und hilft Ihnen, schnellere und effektivere Sicherheitsuntersuchungen durchzuführen. Detective bietet vordefinierte Datenaggregationen, Zusammenfassungen und Kontexte, mit denen Sie Art und Ausmaß möglicher Sicherheitsprobleme analysieren und bewerten können.

Detective extrahiert automatisch zeitbasierte Ereignisse, wie API Anrufe von AWS CloudTrail und Amazon VPC Flow Logs für Ihre AWS Ressourcen schätzen. Es nimmt auch von Amazon GuardDuty generierte Ergebnisse auf. Detective verwendet dann Machine Learning, statistische

Analysen und die Graphentheorie, um Visualisierungen zu erstellen, mit denen Sie effektive Sicherheitsuntersuchungen schneller durchführen können.

Diese Visualisierungen bieten eine einheitliche, interaktive Ansicht des Ressourcenverhaltens und der Interaktionen zwischen ihnen im Zeitverlauf. Anhand dieses Verhaltensdiagramms können Sie potenziell böswillige Aktionen untersuchen, wie z. B. fehlgeschlagene Anmeldeversuche oder verdächtige API Anrufe. Sie können auch sehen, wie sich diese Aktionen auf Ressourcen wie S3-Buckets und -Objekte auswirken.

Weitere Informationen finden Sie im [Administratorhandbuch für Amazon Detective](#).

IAM Access Analyzer

AWS Identity and Access Management Access Analyzer (IAMAccess Analyzer) kann Ihnen helfen, Ressourcen zu identifizieren, die mit einer externen Entität gemeinsam genutzt werden. Sie können IAM Access Analyzer auch verwenden, um IAM Richtlinien anhand der Richtliniengrammatik und bewährten Methoden zu überprüfen und IAM Richtlinien auf der Grundlage der Zugriffsaktivitäten in Ihrem AWS CloudTrail Protokolle.

IAMAccess Analyzer verwendet logisches Denken, um die Ressourcenrichtlinien in Ihrem zu analysieren AWS Umgebung, wie z. B. Bucket-Richtlinien. Mit IAM Access Analyzer for S3 erhalten Sie eine Warnung, wenn ein S3-Bucket so konfiguriert ist, dass er jedem Zugriff auf das Internet oder andere Benutzer ermöglicht AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation. IAMAccess Analyzer for S3 kann beispielsweise melden, dass für einen Bucket Lese- oder Schreibzugriff über eine Bucket-Zugriffskontrollliste (ACL), eine Bucket-Richtlinie, eine Multi-Region-Access-Point-Richtlinie oder eine Access-Point-Richtlinie gewährt wird. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Erkenntnisse, die die Quelle und die Ebene des öffentlichen oder freigegebenen Zugriffs angeben. Anhand dieser Erkenntnisse können Sie sofortige und präzise Korrekturmaßnahmen ergreifen, um den Bucket-Zugriff wie beabsichtigt wiederherzustellen.

Weitere Informationen finden Sie unter [Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3](#).

Amazon Macie

Amazon Macie ist ein Datensicherheitsservice, der sensible Daten mithilfe von Machine Learning und Musterabgleich entdeckt, Einblicke in Datensicherheitsrisiken bietet und automatischen Schutz vor diesen Risiken ermöglicht.

Mit Macie können Sie die Erkennung und Meldung sensibler Daten in Ihren S3-Buckets automatisieren, um die Daten, die Ihre Organisation in Amazon S3 speichert, besser zu verstehen. Um sensible Daten zu erkennen, können Sie die von Macie bereitgestellten integrierten Kriterien und Techniken, benutzerdefinierte Kriterien, die Sie definieren, oder eine Kombination aus beiden verwenden. Wenn Macie sensible Daten in einem S3-Objekt entdeckt, generiert es eine Erkenntnis, um Sie zu benachrichtigen. Diese Erkenntnis enthält Informationen über den betroffenen Bucket und das entsprechende Objekt, die Art und Häufigkeit der Vorkommen der sensiblen Daten, die Macie gefunden hat, sowie zusätzliche Details, die Ihnen die Untersuchung erleichtern.

Macie stellt außerdem Statistiken und andere Daten zur Verfügung, die einen Einblick in die Sicherheitslage Ihrer Amazon-S3-Daten geben, wertet Ihre S3-Buckets automatisch aus und überwacht sie, um die Sicherheit und den Zugriff zu kontrollieren. Wenn Macie ein potenzielles Problem mit der Sicherheit oder dem Datenschutz erkennt, wie einen Bucket, der öffentlich zugänglich wird, generiert Macie eine Erkenntnis, die Sie überprüfen und bei Bedarf korrigieren können.

Weitere Informationen finden Sie im [Benutzerhandbuch für Amazon Macie](#).

AWS Security Hub

AWS Security Hub ist ein Service zur Verwaltung des Sicherheitsstatus, der bewährte Sicherheitsverfahren überprüft, Warnmeldungen und Ergebnisse aus mehreren Quellen in einem einzigen Format zusammenfasst und automatische Problembhebungen ermöglicht.

Security Hub sammelt und liefert Daten zu Sicherheitsergebnissen aus integrierten AWS Partner Network Sicherheitslösungen und AWS-Services, einschließlich Amazon Detective, Amazon GuardDuty, IAM Access Analyzer und Amazon Macie. Es generiert auch seine eigenen Ergebnisse, indem es kontinuierliche, automatisierte Sicherheitsprüfungen durchführt, die auf folgenden Kriterien basieren AWS bewährte Verfahren und unterstützte Industriestandards.

Security Hub korreliert und konsolidiert die Erkenntnisse dann anbieterübergreifend, um Ihnen dabei zu helfen, die wichtigsten Erkenntnisse zu priorisieren und zu verarbeiten. Es bietet auch Unterstützung für benutzerdefinierte Aktionen, mit denen Sie Antworten oder Abhilfemaßnahmen für bestimmte Klassen von Erkenntnissen aufrufen können.

Mit Security Hub können Sie den Sicherheits- und Compliance-Status Ihrer Amazon S3 S3-Ressourcen beurteilen, und zwar im Rahmen einer umfassenderen Analyse der individuellen Sicherheitslage Ihres Unternehmens AWS-Regionen und über mehrere Regionen hinweg. Dazu gehören die Analyse von Sicherheitstrends und die Identifizierung der Sicherheitsprobleme mit der

höchsten Priorität. Sie können auch Ergebnisse aus mehreren zusammenfassenden AWS-Regionen und aggregierte Ergebnisdaten aus einer einzigen Region überwachen und verarbeiten.

Weitere Informationen finden Sie unter [Amazon Simple Storage Service Controls](#) im AWS Security Hub Benutzerleitfaden.

Verwalten Ihres Amazon-S3-Speichers

Nachdem Sie Buckets erstellt und Objekte in Amazon S3 hochgeladen haben, können Sie Ihren Objektspeicher mithilfe von Funktionen wie Versioning, Speicherklassen, Objektsperre, Batch-Vorgänge, Replikation, Markierungen und mehr verwalten. Die folgenden Abschnitte enthalten detaillierte Informationen zu den Speicherverwaltungsfunktionen und Funktionen, die in Amazon S3 verfügbar sind.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Verwenden der Versioning in S3-Buckets](#)
- [Verwenden von AWS Backup für Amazon S3](#)
- [Arbeiten mit archivierten Objekten](#)
- [Verwenden der S3-Objektsperre](#)
- [Verwenden von Amazon-S3-Speicherklassen](#)
- [Langfristige Datenspeicherung mit S3 Glacier-Speicherklassen](#)
- [Amazon S3 Intelligent Tiering](#)
- [Verwaltung des Lebenszyklus von Objekten](#)
- [Amazon S3 Inventory](#)
- [Überblick über das Replizieren von Objekten](#)
- [Kategorisieren des Speichers mithilfe von Markierungen](#)
- [Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets](#)
- [Fakturierung und Nutzungsberichte für Amazon S3](#)
- [Filtern und Abrufen von Daten mit Amazon S3 Select](#)
- [Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch.](#)

Verwenden der Versioning in S3-Buckets

Das Versioning in Amazon S3 ermöglicht Ihnen, mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Sie können die S3-Versioning-Funktion verwenden, um sämtliche Versionen aller Objekte in Ihren Buckets zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leichter wiederherstellen. Wenn Sie das Versioning für einen Bucket aktivieren und Amazon S3 mehrere Schreib Anforderungen für dasselbe Objekt gleichzeitig empfängt, werden alle Objekte gespeichert.

Versioning-fähige Buckets erlauben Ihnen, Objekte nach einem versehentlichen Löschen oder Überschreiben wiederherzustellen. Wenn Sie beispielsweise ein Objekt löschen, fügt Amazon S3 eine Löschmarkierung ein, anstatt das Objekt dauerhaft zu entfernen. Die Löschmarkierung wird zur aktuellen Objektversion. Wenn Sie ein Objekt überschreiben, entsteht eine neue Objektversion im Bucket. Sie können die vorherige Version immer wieder herstellen. Weitere Informationen finden Sie unter [Löschen von Objekten aus einem versioning-fähigen Bucket](#).

Standardmäßig ist das S3-Versioning in Buckets deaktiviert und Sie müssen es explizit aktivieren. Weitere Informationen finden Sie unter [Aktivieren des Versioning für Buckets](#).

Note

- Die unterstützt SOAP API keine S3-Versionierung. SOAPSupport Over HTTP ist veraltet, aber es ist immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstützt SOAP.
- Für jede Version eines gespeicherten und übertragenen Objekts gelten die Standardpreise von Amazon S3. Jede Version eines Objekts besteht aus dem vollständigen Objekt, nicht nur aus einem Delta gegenüber der vorherigen Version. Wenn Sie also drei Versionen eines Objekts gespeichert haben, fallen Gebühren für die drei Objekte an.

Nicht versionierte, versionings-fähige und Buckets mit ausgesetztem Versioning

Buckets können einen von drei Zuständen aufweisen:

- Nicht versioniert (Standard)
- Versioningsfähig

• Ausgesetztes Versioning

Sie aktivieren und unterbrechen das Versioning auf Bucket-Ebene. Nachdem Sie einen Bucket versionsfähig gemacht haben, kann er nicht in einen nicht versionsfähigen Status zurückgesetzt werden. Sie können das Versioning für diesen Bucket jedoch aussetzen.

Der Versioning-Status gilt für alle (niemals für eine Untermenge) der Objekte in diesem Bucket. Wenn Sie die Versioning in einem Bucket aktivieren, werden alle neuen Objekte versioniert und mit einer eindeutigen Versions-ID versehen. Objekte, die zum Zeitpunkt der Aktivierung der Versioning im Bucket bereits im Bucket vorhanden waren, werden immer versioniert und erhalten eine eindeutige Versions-ID, wenn sie durch zukünftige Anforderungen geändert werden. Beachten Sie Folgendes:

- Objekte, die in Ihrem Bucket gespeichert waren, bevor Sie den Versioning-Status einrichten, haben die Versions-ID null. Wenn Sie das Versioning aktivieren, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht mehr. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet. Weitere Informationen finden Sie unter [Arbeiten mit Objekten in einem versioning-fähigen Bucket](#).
- Der Bucket-Eigentümer (oder ein anderer Benutzer mit geeigneten Berechtigungen) kann das Versioning aussetzen, um zu verhindern, dass sich weitere Objektversionen ansammeln. Wenn Sie das Versioning aussetzen, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet. Weitere Informationen finden Sie unter [Arbeiten mit Objekten in einem Bucket mit ausgesetztem Versioning](#).

Verwenden des S3-Versioning mit dem S3-Lebenszyklus

Zum Anpassen Ihrer Datenaufbewahrungsmethode und zur Speicherkostenkontrolle verwenden Sie das Objekt-Versioning mit dem S3-Lebenszyklus. Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#). Informationen zum Erstellen von S3-Lifecycle-Konfigurationen mit dem AWS Management Console, AWS CLI, AWS SDKs, oder das RESTAPI, siehe [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#).

Important

Wenn in dem nicht-versionierten Bucket eine Lebenszyklus-Konfiguration für den Ablauf von Objekten vorhanden ist und Sie dasselbe Verhalten hinsichtlich einer dauerhaften Löschung beim Aktivieren der Versionsverwaltung beibehalten möchten, müssen Sie eine Konfiguration für den Ablauf von nicht aktuellen Objekten hinzufügen. Die Konfiguration

für den Lebenszyklus mit nicht aktuellem Ablauf verwaltet die Löschungen der veralteten Objektversionen im Bucket mit aktivierter Versionierung. (Ein Bucket mit aktivierter Versionierung verwaltet eine aktuelle und keine oder mehrere nicht aktuelle Objektversionen.) Weitere Informationen finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#).

Informationen zum Arbeiten mit S3-Versioning finden Sie in den folgenden Themen.

Themen

- [Wie funktioniert S3-Versioning](#)
- [Aktivieren des Versioning für Buckets](#)
- [MFALöschen konfigurieren](#)
- [Arbeiten mit Objekten in einem versioning-fähigen Bucket](#)
- [Arbeiten mit Objekten in einem Bucket mit ausgesetztem Versioning](#)

Wie funktioniert S3-Versioning

Sie können die S3-Versionsverwaltung verwenden, um mehrere Versionen eines Objekts in einem Bucket zu behalten und so Objekte wiederherzustellen, die versehentlich gelöscht oder überschrieben wurden. Wenn Sie beispielsweise die S3-Versionsverwaltung auf einen Bucket anwenden, werden die folgenden Änderungen vorgenommen:

- Wenn Sie ein Objekt löschen, fügt Amazon S3 eine Löschmarkierung hinzu, statt das Objekt dauerhaft zu entfernen. Die Löschmarkierung wird zur aktuellen Objektversion. Sie können dann die vorherige Version wiederherstellen. Weitere Informationen finden Sie unter [Löschen von Objekten aus einem versioning-fähigen Bucket](#).
- Wenn Sie ein Objekt überschreiben, fügt Amazon S3 eine neue Objektversion im Bucket hinzu. Die vorherige Version verbleibt im Bucket und wird zu einer nicht aktuellen Version. Sie können die vorherige Version wiederherstellen.

Note

Für jede Version eines gespeicherten und übertragenen Objekts gelten die Standardpreise von Amazon S3. Jede Version eines Objekts besteht aus dem vollständigen Objekt, nicht aus

einem Delta gegenüber der vorherigen Version. Wenn Sie also drei Versionen eines Objekts gespeichert haben, fallen Gebühren für die drei Objekte an.

Jedem von Ihnen erstellten S3-Bucket ist eine Versioning-Subressource zugeordnet. (Weitere Informationen finden Sie unter [Optionen für die Bucket-Konfiguration](#).) Standardmäßig ist Ihr Bucket nicht versioning-fähig, und dementsprechend speichert die Versioning-Subressource eine leere Versioning-Konfiguration wie folgt.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Um die Versionsverwaltung zu aktivieren, können Sie eine Anfrage an Amazon S3 mit einer Versionskonfiguration senden, die einen Enabled-Status enthält.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Um das Versioning auszusetzen, setzen Sie den Statuswert auf Suspended.

Note

Wenn Sie die Versionsverwaltung für einen Bucket zum ersten Mal aktivieren, kann es einen Moment dauern, bis die Änderung vollständig verbreitet ist. Wir empfehlen, dass Sie nach dem Aktivieren der Versionsverwaltung 15 Minuten warten, bevor Sie Schreibvorgänge (PUT oder DELETE) für Objekte im Bucket ausführen.

Der Bucket-Besitzer und alle Autorisierten AWS Identity and Access Management (IAM) Benutzer können die Versionierung aktivieren. Der Bucket-Besitzer ist AWS-Konto der den Bucket erstellt hat. Weitere Informationen zu Berechtigungen finden Sie unter [Identity and Access Management für Amazon S3](#).

Weitere Informationen zur Aktivierung und Deaktivierung der S3-Versionierung finden Sie unter AWS Management Console, AWS Command Line Interface (AWS CLI), oder RESTAPI, siehe. [the section called "Aktivieren des Versioning für Buckets"](#)

Themen

- [Ausführung IDs](#)
- [Versioning-Workflows](#)

Ausführung IDs

Wenn Sie das Versioning für einen Bucket aktivieren, generiert Amazon S3 automatisch eine eindeutige Versions-ID für das Objekt, das gespeichert wird. In einem Bucket können Sie beispielsweise zwei Objekte mit demselben Schlüssel (Objektnamen), aber unterschiedlicher Version haben IDs, z. B. `photo.gif` (Version 111111) und `photo.gif` (Version 121212).

Diagramm, das einen Bucket mit aktivierter Versionierung darstellt, der zwei Objekte mit demselben Schlüssel, aber unterschiedlicher Version enthält. IDs

Jedes Objekt hat eine Versions-ID, unabhängig davon, ob die S3-Versionierung aktiviert ist oder nicht. Wenn die S3-Versionsverwaltung nicht aktiviert ist, legt Amazon S3 den Wert der Versions-ID auf `null` fest. Wenn das S3-Versioning aktiviert ist, weist Amazon S3 dem Objekt einen Versions-ID-Wert zu. Dieser Wert unterscheidet dieses Objekt von anderen Versionen desselben Schlüssels.

Wenn Sie das S3-Versioning für einen vorhandenen Bucket aktivieren, bleiben Objekte, die bereits im Bucket gespeichert sind, unverändert. Ihre Version IDs (`null`), ihr Inhalt und ihre Berechtigungen bleiben unverändert. Nachdem Sie die S3-Versionsverwaltung aktiviert haben, erhält jedes Objekt, das dem Bucket hinzugefügt wird, eine Versions-ID, die sie von anderen Versionen desselben Schlüssels unterscheidet.

Nur Amazon S3 generiert Versions-IDs, und sie können nicht bearbeitet werden. Bei den Versions-IDs handelt es sich um Unicode-kodierte, UTF-8-kodierte, undurchsichtige Zeichenketten, die nicht länger als 1.024 Byte sind. Im Folgenden wird ein Beispiel gezeigt:

```
3sL4kqtJ1cpXroDTDmJ+rmspXd3dIb1HY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

Note

Der Einfachheit halber verwenden die anderen Beispiele in diesem Thema viel kürzere IDs

Versioning-Workflows

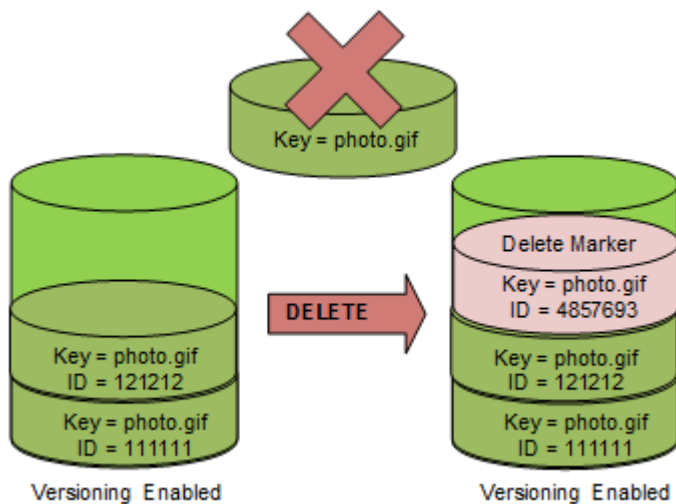
Wenn Sie mit PUT ein Objekt in einen Bucket mit aktiviertem Versioning schreiben, wird die nicht aktuelle Version nicht überschrieben. Wie in der folgenden Abbildung dargestellt, tritt das folgende

Verhalten auf, wenn eine neue Version von `photo.gif` mit PUT in einen Bucket geschrieben wird, der bereits ein Objekt desselben Namens enthält:

- Das ursprüngliche Objekt (ID = 111111) verbleibt im Bucket.
- Amazon S3 generiert eine neue Versions-ID (121212) und fügt diese neuere Version des Objekts dem Bucket hinzu.

Mit dieser Funktion können Sie eine vorherige Version eines Objekts abrufen, wenn ein Objekt versehentlich überschrieben oder gelöscht wurde.

Wenn Sie DELETE für ein Objekt ausführen, bleiben alle Versionen in dem Bucket, und Amazon S3 fügt eine Löschmarkierung ein, wie in der folgenden Abbildung gezeigt.



Die Löschmarkierung wird zur aktuellen Version des Objekts. Standardmäßig wird durch GET-Anforderungen die zuletzt gespeicherte Version abgerufen. Eine GET Object-Anforderung gibt einen 404 Not Found-Fehler zurück, wenn die aktuelle Version eine Löschmarkierung ist, wie in der folgenden Abbildung gezeigt.

Sie können jedoch mit GET eine nicht aktuelle Version eines Objekts abrufen, indem Sie ihre Versions-ID angeben. In der folgenden Abbildung wird mit GET eine spezifische Objektversion abgerufen, 111111. Amazon S3 gibt diese Objektversion zurück, auch wenn es nicht die aktuelle Version ist.

Weitere Informationen finden Sie unter [Abrufen von Objektversionen aus einem versioning-fähigen Bucket](#).

Sie können ein Objekt permanent löschen, indem Sie die Version angeben, die Sie löschen wollen. Nur der Besitzer eines Amazon S3 S3-Buckets oder ein autorisierter IAM Benutzer kann eine Version dauerhaft löschen. Wenn Ihre DELETE-Operation die `versionId` angibt, wird diese Objektversion dauerhaft gelöscht, und Amazon S3 fügt keine Löschmarkierung ein.

Sie können die Sicherheit erhöhen, indem Sie einen Bucket so konfigurieren, dass das Löschen mit Multi-Faktor-Authentifizierung (MFA) aktiviert wird. Wenn Sie das MFA Löschen für einen Bucket aktivieren, muss der Bucket-Besitzer in jeder Anfrage zwei Arten der Authentifizierung angeben, um eine Version zu löschen oder den Versionsstatus des Buckets zu ändern. Weitere Informationen finden Sie unter [MFALöschen konfigurieren](#).

Wann werden neue Versionen für ein Objekt erstellt?

Neue Versionen werden nur erstellt, wenn Sie für ein neues Objekt PUT ausführen. Beachten Sie, dass bestimmte Aktionen wie `CopyObject` durch die Implementierung einer PUT-Operation funktionieren.

Einige Aktionen, die das aktuelle Objekt ändern, erstellen keine neue Version, da sie kein neues Objekt mit PUT bearbeiten. Dies umfasst Aktionen wie das Ändern der Markierungen für ein Objekt.

Important

Wenn Sie einen deutlichen Anstieg der Anzahl von HTTP 503 Antworten (Service Unavailable) für Amazon S3 PUT oder DELETE Objektanfragen an einen Bucket mit aktivierter S3-Versionierung feststellen, haben Sie möglicherweise ein oder mehrere Objekte im Bucket, für die es Millionen von Versionen gibt. Weitere Informationen finden Sie im Abschnitt zur S3-Versionsverwaltung unter [Fehlerbehebung](#).

Aktivieren des Versioning für Buckets

Sie können Amazon-S3-Versioning nutzen, um mehrere Versionen eines Objekts in einem Bucket aufzubewahren. Dieser Abschnitt enthält Beispiele dafür, wie Sie die Versionierung für einen Bucket mithilfe der Konsole aktivieren, REST API AWS SDKs, und AWS Command Line Interface (AWS CLI).

Note

Wenn Sie die Versionierung für einen Bucket zum ersten Mal aktivieren, kann es bis zu 15 Minuten dauern, bis die Änderung vollständig übernommen ist. Wir empfehlen, nach

der Aktivierung der Versionierung 15 Minuten zu warten, bevor Sie Schreibvorgänge (PUT oder DELETE) für Objekte im Bucket ausführen. Schreibvorgänge, die ausgeführt werden, bevor diese Konvertierung abgeschlossen ist, können sich auf Objekte ohne Version beziehen.

Weitere Informationen über das S3-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#). Informationen zum Arbeiten mit Objekten, die sich in einem versioning-fähigen Bucket befinden, finden Sie unter [Arbeiten mit Objekten in einem versioning-fähigen Bucket](#).

Weitere Informationen zur Verwendung von S3 Versioning zum Schutz von Daten finden Sie im [Tutorial: Schutz von Daten in Amazon S3 vor versehentlichem Löschen oder Anwendungsfehlern mithilfe von S3 Versioning, S3 Object Lock und S3 Replication](#).

Jedem von Ihnen erstellten S3-Bucket ist eine Versioning-Subressource zugeordnet. (Weitere Informationen finden Sie unter [Optionen für die Bucket-Konfiguration](#).) Standardmäßig ist Ihr Bucket nicht versioning-fähig, und dementsprechend speichert die Versioning-Subressource eine leere Versioning-Konfiguration wie folgt.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Um das Versioning zu aktivieren, können Sie eine Anfrage an Amazon S3 mit einer Versionskonfiguration senden, die einen Status enthält.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Um das Versioning auszusetzen, setzen Sie den Statuswert auf Suspended.

Der Bucket-Eigentümer und alle autorisierten Benutzer können das Versioning aktivieren. Der Besitzer des Buckets ist AWS-Konto der den Bucket (das Root-Konto) erstellt hat. Weitere Informationen zu Berechtigungen finden Sie unter [Identity and Access Management für Amazon S3](#).

In den folgenden Abschnitten finden Sie weitere Informationen zur Aktivierung der S3-Versionierung über die Konsole, AWS CLI, und die AWS SDKs.

Verwenden der S3-Konsole

Folgen Sie diesen Schritten, um das zu verwenden AWS Management Console um die Versionierung in einem S3-Bucket zu aktivieren.

Aktivieren und Deaktivieren des Versionings für einen S3-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Versioning aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Bucket Versioning (Bucket-Versioning) die Option Edit (Bearbeiten).
5. Wählen Sie Suspend (Anhalten) oder Enable (Aktivieren) und dann Save changes (Änderungen speichern).

Note

Sie können Folgendes verwenden ... AWS Multi-Faktor-Authentifizierung (MFA) mit Versionierung. Wenn Sie MFA mit Versionierung verwenden, müssen Sie Ihre angeben AWS-Konto die Zugriffsschlüssel und ein gültiger Code vom MFA Gerät des Kontos, um eine Objektversion dauerhaft zu löschen oder die Versionierung auszusetzen oder zu reaktivieren. Zur Verwendung MFA zusammen mit der Versionierung aktivieren Sie. MFA Delete Sie können es jedoch nicht MFA Delete mit dem aktivieren AWS Management Console. Sie müssen die verwenden AWS Command Line Interface (AWS CLI) oder dasAPI. Weitere Informationen finden Sie unter [MFALöschen konfigurieren](#).

Verwendung der AWS CLI

Im folgenden Beispiel wird das Versioning auf einem S3-Bucket aktiviert.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-configuration Status=Enabled
```

Das folgende Beispiel aktiviert die S3-Versionierung und das Löschen mit Multi-Faktor-Authentifizierung (MFA) für einen Bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Note

Für die Verwendung von MFA delete ist ein zugelassenes physisches oder virtuelles Authentifizierungsgerät erforderlich. Weitere Informationen zur Verwendung von MFA Delete in Amazon S3 finden Sie unter [MFALöschen konfigurieren](#).

Weitere Informationen zur Aktivierung der Versionierung finden Sie unter AWS CLI, siehe [put-bucket-versioning](#) in der AWS CLI Befehlsreferenz.

Verwendung der AWS SDKs

In den folgenden Beispielen wird die Versionsverwaltung für einen Bucket aktiviert und anschließend der Versionsstatus mit dem Befehl abgerufen AWS SDK for Java und das AWS SDK for .NET. Für Informationen zur Verwendung anderer AWS SDKs, siehe [AWS Entwicklerzentrum](#).

.NET

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using System;
using Amazon.S3;
using Amazon.S3.Model;

namespace s3.amazon.com.docsamples
{
    class BucketVersioningConfiguration
    {
        static string bucketName = "*** bucket name ***";

        public static void Main(string[] args)
        {
```

```
using (var client = new AmazonS3Client(Amazon.RegionEndpoint.USEast1))
{
    try
    {
        EnableVersioningOnBucket(client);
        string bucketVersioningStatus =
RetrieveBucketVersioningConfiguration(client);
    }
    catch (AmazonS3Exception amazonS3Exception)
    {
        if (amazonS3Exception.ErrorCode != null &&
            (amazonS3Exception.ErrorCode.Equals("InvalidAccessKeyId")
            ||
            amazonS3Exception.ErrorCode.Equals("InvalidSecurity")))
        {
            Console.WriteLine("Check the provided AWS Credentials.");
            Console.WriteLine(
                "To sign up for service, go to http://aws.amazon.com/s3");
        }
        else
        {
            Console.WriteLine(
                "Error occurred. Message:'{0}' when listing objects",
                amazonS3Exception.Message);
        }
    }
}

Console.WriteLine("Press any key to continue...");
Console.ReadKey();
}

static void EnableVersioningOnBucket(IAmazonS3 client)
{
    PutBucketVersioningRequest request = new PutBucketVersioningRequest
    {
        BucketName = bucketName,
        VersioningConfig = new S3BucketVersioningConfig
        {
            Status = VersionStatus.Enabled
        }
    };
};
```



```
        PutBucketVersioningResponse response =
client.PutBucketVersioning(request);
    }

    static string RetrieveBucketVersioningConfiguration(IAmazonS3 client)
    {
        GetBucketVersioningRequest request = new GetBucketVersioningRequest
        {
            BucketName = bucketName
        };

        GetBucketVersioningResponse response =
client.GetBucketVersioning(request);
        return response.VersioningConfig.Status;
    }
}
}
```

Java

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;

public class BucketVersioningConfigurationExample {
    public static String bucketName = "**** bucket name ****";
    public static AmazonS3Client s3Client;

    public static void main(String[] args) throws IOException {
        s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
        s3Client.setRegion(Region.getRegion(Regions.US_EAST_1));
        try {
```

```
// 1. Enable versioning on the bucket.
BucketVersioningConfiguration configuration =
    new BucketVersioningConfiguration().withStatus("Enabled");

SetBucketVersioningConfigurationRequest setBucketVersioningConfigurationRequest
=
    new SetBucketVersioningConfigurationRequest(bucketName, configuration);

s3Client.setBucketVersioningConfiguration(setBucketVersioningConfigurationRequest);

// 2. Get bucket versioning configuration information.
BucketVersioningConfiguration conf =
s3Client.getBucketVersioningConfiguration(bucketName);
    System.out.println("bucket versioning configuration status:    " +
conf.getStatus());

        } catch (AmazonS3Exception amazonS3Exception) {
            System.out.format("An Amazon S3 error occurred. Exception: %s",
amazonS3Exception.toString());
        } catch (Exception ex) {
            System.out.format("Exception: %s", ex.toString());
        }
    }
}
```

Python

Das folgende Python-Beispiel erstellt einen Amazon-S3-Bucket, aktiviert ihn für das Versioning und konfiguriert einen Lebenszyklus, der nicht-aktuelle Objektversionen nach 7 Tagen ablaufen lässt.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
    lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.
```

Usage is shown in the `usage_demo_single_object` function at the end of this module.

```
:param bucket_name: The name of the bucket to create.
:param prefix: Identifies which objects are automatically expired under the
               configured lifecycle rules.
:return: The newly created bucket.
"""
try:
    bucket = s3.create_bucket(
        Bucket=bucket_name,
        CreateBucketConfiguration={
            "LocationConstraint": s3.meta.client.meta.region_name
        },
    )
    logger.info("Created bucket %s.", bucket.name)
except ClientError as error:
    if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
        logger.warning("Bucket %s already exists! Using it.", bucket_name)
        bucket = s3.Bucket(bucket_name)
    else:
        logger.exception("Couldn't create bucket %s.", bucket_name)
        raise

try:
    bucket.Versioning().enable()
    logger.info("Enabled versioning on bucket %s.", bucket.name)
except ClientError:
    logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
    raise

try:
    expiration = 7
    bucket.LifecycleConfiguration().put(
        LifecycleConfiguration={
            "Rules": [
                {
                    "Status": "Enabled",
                    "Prefix": prefix,
                    "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration}],
            ]
        }
    )
```

```
)
logger.info(
    "Configured lifecycle to expire noncurrent versions after %s days "
    "on bucket %s.",
    expiration,
    bucket.name,
)
except ClientError as error:
    logger.warning(
        "Couldn't configure lifecycle on bucket %s because %s. "
        "Continuing anyway.",
        bucket.name,
        error,
    )

return bucket
```

MFALöschen konfigurieren

Wenn Sie mit der S3-Versionierung in Amazon S3 S3-Buckets arbeiten, können Sie optional eine weitere Sicherheitsebene hinzufügen, indem Sie einen Bucket so konfigurieren, dass er das Löschen MFA(Multi-Faktor-Authentifizierung) aktiviert. In diesem Fall muss der Bucket-Eigentümer zwei Authentifizierungsformen in jede Anforderung aufnehmen, um eine Version zu löschen oder den Versioning-Status des Buckets zu ändern.

MFADelete erfordert eine zusätzliche Authentifizierung für einen der folgenden Vorgänge:

- Ändern des Versioning-Status Ihres Buckets
- Dauerhaftes Löschen einer Objektversion

MFADelete erfordert zwei Arten der Authentifizierung zusammen:

- Ihre Sicherheitsanmeldeinformationen
- Die Verkettung einer gültigen Seriennummer, eines Leerzeichens und des sechsstelligen Codes, der auf einem zugelassenen Authentifizierungsgerät angezeigt wird

MFAdelete bietet somit zusätzliche Sicherheit, falls beispielsweise Ihre Sicherheitsanmeldedaten kompromittiert wurden. MFAdelete kann dazu beitragen, das versehentliche Löschen von Buckets zu verhindern, indem der Benutzer, der die Löschaktion initiiert, den physischen Besitz eines MFA Geräts anhand eines MFA Codes nachweisen muss. Dadurch wird die Löschaktion noch reibungsloser und sicherer.

Um Buckets zu identifizieren, für die das MFA Löschen aktiviert ist, können Sie Amazon S3 Storage Lens-Metriken verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter [Bewertung Ihrer Speicheraktivität und -nutzung mit S3 Storage Lens](#). Eine vollständige Liste der Metriken finden Sie im [Glossar der S3-Storage-Lens-Metriken](#).

Der Bucket-Besitzer, der AWS-Konto der den Bucket erstellt hat (Root-Konto), und alle autorisierten Benutzer können die Versionierung aktivieren. Allerdings kann nur der Bucket-Besitzer (Root-Konto) das MFA Löschen aktivieren. Weitere Informationen finden Sie unter [Sichern des Zugriffs auf AWS Verwenden MFA](#) auf dem AWS Sicherheits-Blog.

Note

Um MFA Löschen zusammen mit der Versionierung zu verwenden, aktivieren MFA Delete Sie. Sie können es jedoch nicht MFA Delete mit dem aktivieren AWS Management Console. Sie müssen die verwenden AWS Command Line Interface (AWS CLI) oder dasAPI. Beispiele für die Verwendung von MFA Delete zusammen mit der Versionierung finden Sie im Abschnitt mit den Beispielen des Themas [Aktivieren des Versioning für Buckets](#). Sie können MFA Delete nicht mit Lebenszykluskonfigurationen verwenden. Weitere Informationen zu Lebenszykluskonfigurationen und deren Interaktion mit anderen Konfigurationen finden Sie unter [Lebenszyklus- und andere Bucket-Konfigurationen](#).

Um das MFA Löschen zu aktivieren oder zu deaktivieren, verwenden Sie dasselbe VerfahrenAPI, mit dem Sie die Versionierung für einen Bucket konfigurieren. Amazon S3 speichert die MFA Löschkonfiguration in derselben Versionierungs-Subressource, die den Versionsstatus des Buckets speichert.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
```

```
</VersioningConfiguration>
```

Um MFA Delete zu verwenden, können Sie entweder ein Hardware- oder ein virtuelles MFA Gerät verwenden, um einen Authentifizierungscode zu generieren. Das folgende Beispiel zeigt einen generierten Authentifizierungscode, angezeigt auf einem Hardwaregerät.



MFALöschen und MFA geschützter API Zugriff sind Funktionen, die Schutz für verschiedene Szenarien bieten sollen. Sie konfigurieren das MFA Löschen für einen Bucket, um sicherzustellen, dass die Daten in Ihrem Bucket nicht versehentlich gelöscht werden können. MFA-geschützter API Zugriff wird verwendet, um einen anderen Authentifizierungsfaktor (MFACode) beim Zugriff auf sensible Amazon S3 S3-Ressourcen durchzusetzen. Sie können verlangen, dass alle Operationen mit diesen Amazon S3 S3-Ressourcen mit temporären Anmeldeinformationen ausgeführt werden, die mit erstellt wurden MFA. Ein Beispiel finden Sie unter [Erforderlich MFA](#).

Weitere Informationen zum Kauf und zur Aktivierung eines Authentifizierungsgeräts finden Sie unter [Multi-Faktor-Authentifizierung](#).

Um die S3-Versionierung zu aktivieren und zu konfigurieren MFA, löschen

Verwendung der AWS CLI

Das folgende Beispiel aktiviert die S3-Versionierung und das Löschen mit Multi-Faktor-Authentifizierung (MFA) für einen Bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Mit dem REST API

Weitere Informationen zur Angabe von MFA Delete mithilfe von Amazon S3 REST API finden Sie unter [PutBucketVersioning](#) Amazon Simple Storage Service API Reference.

Arbeiten mit Objekten in einem versioning-fähigen Bucket

Objekte, die in einem Amazon-S3-Bucket gespeichert waren, bevor Sie den Versioning-Status einrichten, haben die Versions-ID null. Wenn Sie das Versioning aktivieren, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht mehr. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet.

Übergang von Objektversionen

Sie können Lebenszyklus-Konfigurationsregeln für Objekte definieren, die einen definierten Lebenszyklus haben, um Objektversionen zu einem bestimmten Zeitpunkt in der Lebensdauer des Objekts in die Speicherklasse S3 Glacier Flexible Retrieval zu überführen. Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Die Themen in diesem Abschnitt erklären verschiedene Objekt-Vorgänge in einem Bucket mit aktiviertem Versioning. Weitere Informationen über das Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

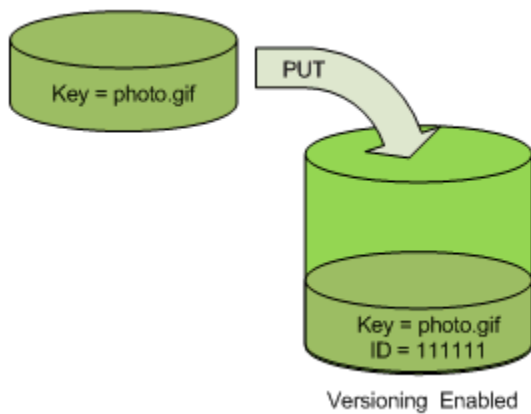
Themen

- [Hinzufügen von Objekten zu versioning-fähigen Buckets](#)
- [Auflisten von Objekten in einem versioning-fähigen Bucket](#)
- [Abrufen von Objektversionen aus einem versioning-fähigen Bucket](#)
- [Löschen von Objekten aus einem versioning-fähigen Bucket](#)
- [Konfigurieren von versionierten Objektberechtigungen](#)

Hinzufügen von Objekten zu versioning-fähigen Buckets

Nachdem Sie das Versioning für einen Bucket aktiviert haben, fügt Amazon S3 jedem im Bucket gespeicherten Objekt automatisch eine eindeutige Versions-ID hinzu (mit PUT, POST oder CopyObject).

Die folgende Abbildung zeigt, dass Amazon S3 jedem Objekt automatisch eine eindeutige Versions-ID hinzufügt, wenn es einem versioningfähigen Bucket hinzugefügt wird.



Note

Die von Amazon S3 zugewiesenen Versions-ID-Werte sind URL sicher (können als Teil einer enthalten seinURI).

Weitere Informationen über das Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#). Sie können mithilfe der Konsole Objektversionen zu einem Bucket mit aktivierter Versionierung hinzufügen. AWS SDKs, und. REST API

Verwenden der Konsole

Detaillierte Anweisungen finden Sie unter [Objekte hochladen](#).

Verwendung der AWS SDKs

Beispiele für das Hochladen von Objekten mit dem AWS SDKs für Java, .NET, und PHP, siehe [Objekte hochladen](#). Die Beispiele für das Hochladen von Objekten in versioning-fähige und nicht versioning-fähige Buckets sind gleich, aber bei versioning-fähigen Buckets weist Amazon S3 eine Versionsnummer zu. Andernfalls ist die Versionsnummer null.

Informationen zur Verwendung anderer AWS SDKs, siehe [AWS Entwicklerzentrum](#).

Mit dem REST API

Hinzufügen von Objekten zu versioning-fähigen Buckets

1. Aktivieren des Versionings für einen Bucket mit einer `PutBucketVersioning`-Anforderung.

Weitere Informationen finden Sie unter [PutBucketVersioning](#) in der Amazon Simple Storage Service API Reference.

2. Senden Sie eine PUT-, POST- oder CopyObject-Anforderung, um ein Objekt im Bucket zu speichern.

Wenn Sie einem versioning-fähigen Bucket ein Objekt hinzufügen, gibt Amazon S3 die Versions-ID des Objekts im Antwort-Header `x-amz-version-id` zurück, wie im folgenden Beispiel gezeigt.

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

Auflisten von Objekten in einem versioning-fähigen Bucket

Dieser Abschnitt zeigt Beispiele, wie Objektversionen aus einem versioning-fähigen Bucket aufgelistet werden. Amazon S3 speichert Versionsinformationen zu Objekten in der versions-Subressource, die dem Bucket zugeordnet ist. Weitere Informationen finden Sie unter [Optionen für die Bucket-Konfiguration](#). Um die Objekte in einem versionsfähigen Bucket aufzulisten, benötigen Sie die `ListBucketVersions`-Berechtigung.

Verwenden der S3-Konsole

Befolgen Sie diese Schritte, um die Amazon-S3-Konsole zu verwenden, um die verschiedenen Versionen eines Objekts anzuzeigen.


Mehrere Versionen eines Objekts anzeigen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Um eine Liste der Versionen der Objekte in dem Bucket anzuzeigen, wählen Sie den Schalter Show versions (Versionen anzeigen).


Die Konsole zeigt für jede Objektversion eine eindeutige Versions-ID, das Datum und die Uhrzeit, wann das Objekt erstellt wurde, sowie weitere Eigenschaften an. (Objekte, die in Ihrem Bucket gespeichert waren, bevor Sie den Versioning-Status einrichten, haben die Versions-ID null.)

Um die Objekte ohne die Versionen aufzulisten, wählen Sie den Schalter List versions (Versionen auflisten) .

Objektversionen können auch in der Objektübersicht auf der Konsole angezeigt, heruntergeladen und gelöscht werden. Weitere Informationen finden Sie unter [Anzeigen einer Objektübersicht in der Amazon-S3-Konsole](#).

 Note

Um auf Objektversionen zuzugreifen, die älter als 300 Versionen sind, müssen Sie die AWS CLI oder die des ObjektsURL.


 Important

Sie können den Löschvorgang für ein Objekt nur rückgängig machen, wenn seine aktuelle Version gelöscht wurde. Es ist nicht möglich, das Löschen einer vorherigen Version eines Objekts rückgängig zu machen, das gelöscht wurde. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Verwendung der AWS SDKs

Die Beispiele in diesem Abschnitt veranschaulichen, wie Sie eine Objektliste aus einem versioning-fähigen Bucket abrufen. Jede Anforderung gibt bis zu 1000 Versionen zurück, sofern Sie keine kleinere Anzahl festlegen. Wenn die Versionen im Bucket dieses Limit überschreiten, senden Sie eine Reihe von Anforderungen, um die Liste aller Versionen abzurufen. Dieser Vorgang zur „seitenweisen“ Rückgabe von Ergebnissen wird als Paginierung bezeichnet.

Um zu veranschaulichen, wie Paginierung funktioniert, limitieren die Beispiele jede Antwort auf zwei Objektversionen. Nachdem die erste Seite mit Ergebnissen abgerufen wurde, wird in jedem Beispiel überprüft, ob die Versionsliste abgeschnitten wurde. Wurde sie abgeschnitten, dann wird im Beispiel mit dem seitenweisen Abruf fortgefahren, bis alle Versionen abgerufen wurden.

 Note

Die folgenden Beispiele funktionieren auch bei einem Bucket, der versioning-fähig ist, oder bei Objekten ohne individuelle Versionen. In solchen Fällen gibt Amazon S3 die Objektliste mit der Versions-ID von `null` zurück.

Für Informationen über die Verwendung anderer AWS SDKs, siehe [AWS Entwicklerzentrum](#).

Java

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListVersionsRequest;
import com.amazonaws.services.s3.model.S3VersionSummary;
import com.amazonaws.services.s3.model.VersionListing;

public class ListKeysVersioningEnabledBucket {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            // Retrieve the list of versions. If the bucket contains more versions
            // than the specified maximum number of results, Amazon S3 returns
            // one page of results per request.
            ListVersionsRequest request = new ListVersionsRequest()
                .withBucketName(bucketName)
                .withMaxResults(2);
            VersionListing versionListing = s3Client.listVersions(request);
            int numVersions = 0, numPages = 0;
            while (true) {
                numPages++;
                for (S3VersionSummary objectSummary :
versionListing.getVersionSummaries()) {
                    System.out.printf("Retrieved object %s, version %s\n",
                        objectSummary.getKey(),
                        objectSummary.getVersionId());
                    numVersions++;
                }
            }
        }
    }
}
```

```
        }
        // Check whether there are more pages of versions to retrieve. If
        // there are, retrieve them. Otherwise, exit the loop.
        if (versionListing.isTruncated()) {
            versionListing =
s3Client.listNextBatchOfVersions(versionListing);
        } else {
            break;
        }
    }
    System.out.println(numVersions + " object versions retrieved in " +
numPages + " pages");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

.NET

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ListObjectsVersioningEnabledBucketTest
    {
        static string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
```

```
private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
private static IAmazonS3 s3Client;

public static void Main(string[] args)
{
    s3Client = new AmazonS3Client(bucketRegion);
    GetObjectListWithAllVersionsAsync().Wait();
}

static async Task GetObjectListWithAllVersionsAsync()
{
    try
    {
        ListVersionsRequest request = new ListVersionsRequest()
        {
            BucketName = bucketName,
            // You can optionally specify key name prefix in the request
            // if you want list of object versions of a specific object.

            // For this example we limit response to return list of 2
versions.
            MaxKeys = 2
        };
        do
        {
            ListVersionsResponse response = await
s3Client.ListVersionsAsync(request);
            // Process response.
            foreach (S3ObjectVersion entry in response.Versions)
            {
                Console.WriteLine("key = {0} size = {1}",
                    entry.Key, entry.Size);
            }

            // If response is truncated, set the marker to get the next
            // set of keys.
            if (response.IsTruncated)
            {
                request.KeyMarker = response.NextKeyMarker;
                request.VersionIdMarker = response.NextVersionIdMarker;
            }
        }
        else
        {

```

```
        request = null;
    }
    } while (request != null);
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
}
}
}
```

Mit dem REST API

Example – Alle Objektversionen in einem Bucket auflisten

Um alle Versionen aller Objekte in einem Bucket aufzulisten, verwenden Sie die `versions`-Subressource in einer GET Bucket-Anfrage. Amazon S3 kann maximal 1000 Objekte abrufen. Jede Objektversion zählt als vollständiges Objekt. Wenn ein Bucket also zwei Schlüssel enthält (z. B. `photo.gif` und `picture.jpg`) und der erste Schlüssel 990 Versionen und der zweite Schlüssel 400 Versionen hat, ruft eine einzelne Abfrage alle 990 Versionen von `photo.gif` und nur die 10 neuesten Versionen von `picture.jpg` ab.

Amazon S3 gibt Objektversionen in der Reihenfolge zurück, in der sie gespeichert wurden, wobei die zuletzt gespeicherte zuerst zurückgegeben wird.

Geben Sie in einer GET Bucket-Anforderung die `versions`-Subressource an.

```
GET /?versions HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Example – Abruf aller Versionen eines Schlüssels

Wenn Sie eine Untermenge von Objektversionen abrufen möchten, verwenden Sie die Anforderungsparameter für `GET Bucket`. Weitere Informationen finden Sie unter [GET Bucket](#).

1. Setzen Sie den Parameter `prefix` auf den Schlüssel des Objekts, das Sie abrufen wollen.
2. Senden Sie eine `GET Bucket`-Anforderung unter Verwendung der `versions`-Subressource und `prefix`.

```
GET /?versions&prefix=objectName HTTP/1.1
```

Example – Abrufen von Objekten unter Verwendung eines Präfix

Das folgende Beispiel ruft Objekte ab, deren Schlüssel ist oder damit beginnt `myObject`.

```
GET /?versions&prefix=myObject HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Sie können die anderen Anforderungsparameter verwenden, um eine Untermenge aller Versionen des Objekts abzurufen. Weitere Informationen finden Sie unter [GET Bucket](#) in der Amazon Simple Storage Service API Reference.

Example – Abrufen einer Liste zusätzlicher Objekte, falls die Antwort gekürzt wurde

Wenn die Anzahl der Objekte, die in einer `GET`-Anforderung zurückgegeben werden können, den Wert `max-keys` überschreitet, enthält die Antwort `<isTruncated>true</isTruncated>`, ebenso wie den ersten Schlüssel (in `NextKeyMarker`) und die erste Versions-ID (in `NextVersionIdMarker`), die die Anforderungskriterien erfüllen, aber nicht zurückgegeben wurden. Diese zurückgegebenen Werte verwenden Sie als Ausgangspunkt in einer nachfolgenden Anforderung, um die zusätzlichen Objekte abzurufen, die die `GET`-Anforderung erfüllen.

Gehen Sie wie folgt vor, um zusätzlichen Objekte abzurufen, die die ursprüngliche `GET Bucket versions`-Anforderung von einem Bucket erfüllen. Weitere Informationen zu `key-marker`, `version-id-marker`, `NextKeyMarker`, `NextVersionIdMarker`, und finden Sie unter [GET Bucket](#) in der Amazon Simple Storage Service API Reference.

Im Folgenden finden Sie zusätzliche Antworten, die die ursprüngliche `GET`-Anforderung erfüllen:

- Setzen Sie den Wert von `key-marker` auf den Schlüssel, der in `NextKeyMarker` in der vorherigen Antwort zurückgegeben wurde.
- Setzen Sie den Wert von `version-id-marker` auf die Versions-ID, die in `NextVersionIdMarker` in der vorherigen Antwort zurückgegeben wurde.
- Senden Sie eine `GET Bucket versions`-Anforderung mit `key-marker` und `version-id-marker`.

Example – Abrufen von Objekten ab einem bestimmten Schlüssel und einer bestimmten Versions-ID

```
GET /?versions&key-marker=myObject&version-id-marker=298459348571 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Verwendung der AWS CLI

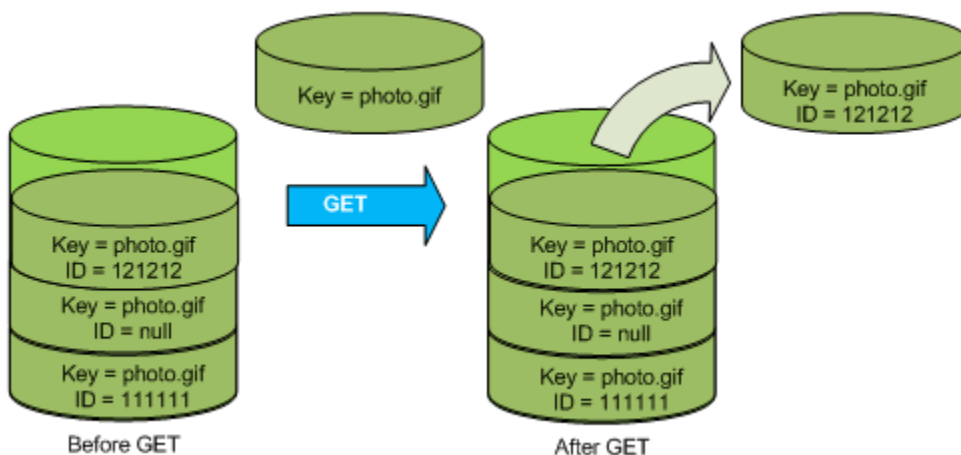
Der folgende Befehl gibt Metadaten zu allen Versionen der Objekte in einem Bucket zurück.

```
aws s3api list-object-versions --bucket amzn-s3-demo-bucket1
```

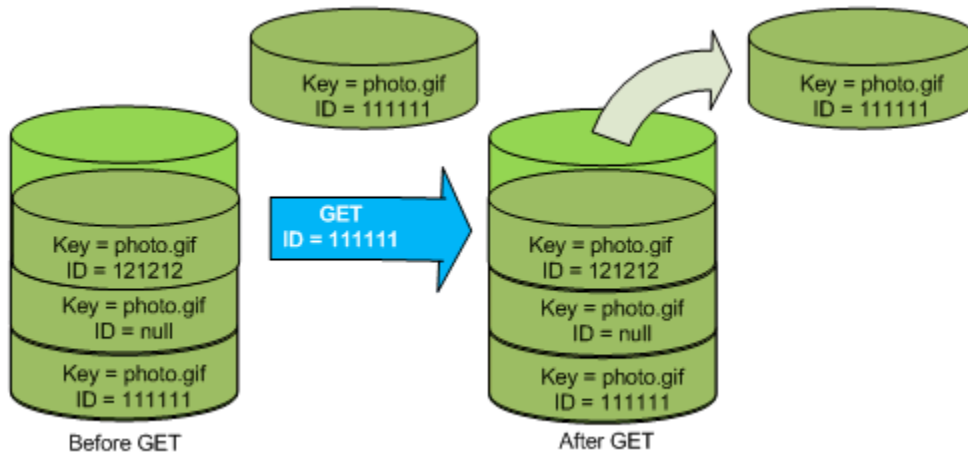
Weitere Informationen zu finden `list-object-versions` Sie unter [list-object-versions](#) in der AWS CLI Befehlsreferenz.

Abrufen von Objektversionen aus einem versioning-fähigen Bucket

Das Versioning in Amazon S3 ist eine Möglichkeit, mehrere Varianten eines Objekts im selben Bucket zu behalten. Eine einfache `GET`-Anforderung ruft die aktuelle Version eines Objekts ab. Die folgende Abbildung zeigt, wie `GET` die aktuelle Version des Objekts zurückgibt, `photo.gif`.



Um eine spezifische Version abzurufen, müssen Sie ihre Versions-ID angeben. Die folgende Abbildung zeigt, dass eine GET `versionId`-Anforderung die angegebene Version des Objekts zurückgibt (nicht unbedingt die aktuelle).



Sie können Objektversionen in Amazon S3 mithilfe der Konsole abrufen. AWS SDKs, oder RESTAPI.

Note

Um auf Objektversionen zuzugreifen, die älter als 300 Versionen sind, müssen Sie die AWS CLI oder die des ObjektsURL.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus.
4. Wählen Sie Versions (Versionen).

Amazon S3 zeigt alle Versionen für das Objekt an.

5. Aktivieren Sie das Kontrollkästchen neben der Version ID (Versions-ID) für die Versionen, die Sie abrufen möchten.
6. Wählen Sie Actions (Aktionen), wählen Sie Download (Herunterladen) und speichern Sie das Objekt.

Objektversionen können auch in der Objektübersicht angezeigt, heruntergeladen und gelöscht werden. Weitere Informationen finden Sie unter [Anzeigen einer Objektübersicht in der Amazon-S3-Konsole](#).

Important

Sie können den Löschvorgang für ein Objekt nur rückgängig machen, wenn seine aktuelle Version gelöscht wurde. Es ist nicht möglich, das Löschen einer vorherigen Version eines Objekts rückgängig zu machen, das gelöscht wurde. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Verwendung der AWS SDKs

Die Beispiele für das Hochladen von Objekten in nicht versioning-fähigen und versioning-fähigen Buckets sind dieselben. Für versioning-fähige Buckets weist Amazon S3 jedoch eine Versionsnummer zu. Andernfalls ist die Versionsnummer null.

Beispiele für das Herunterladen von Objekten mit AWS SDKs für Java, .NET, und PHP, siehe [Objekte herunterladen](#).

Beispiele für das Auflisten der Version von Objekten finden Sie mit AWS SDKs für .NET und Rust, siehe [Auflisten der Version von Objekten in einem Amazon S3 S3-Bucket](#).

Mit dem REST API

Abrufen einer spezifischen Objektversion

1. Setzen Sie den Parameter `versionId` auf die ID der Version des Objekts, die Sie abrufen wollen.
2. Senden Sie eine `GET Object versionId`-Anforderung.

Example – Abrufen eines versionierten Objekts

Die folgende Anforderung ruft die Version `L4kqtJlcpXroDTDmpUMLUo` von `ab my-image.jpg`.

```
GET /my-image.jpg?versionId=L4kqtJlcpXroDTDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Sie können nur die Metadaten eines Objekts (nicht den Inhalt) abrufen. Weitere Informationen finden Sie unter [the section called "Abrufen von Versionsmetadaten"](#).

Informationen zum Wiederherstellen einer früheren Objektversion finden Sie unter [the section called "Wiederherstellen früherer Versionen"](#).

Abrufen der Metadaten einer Objektversion

Wenn Sie nur die Metadaten eines Objekts abrufen wollen (nicht seinen Inhalt), verwenden Sie die HEAD-Operation. Standardmäßig erhalten Sie die Metadaten der aktuellsten Version. Um die Metadaten einer spezifischen Objektversion abzurufen, müssen Sie ihre Versions-ID angeben.

Abruf der Metadaten einer Objektversion

1. Setzen Sie den Parameter `versionId` auf die ID der Version des Objekts, dessen Metadaten Sie abrufen wollen.
2. Senden Sie eine HEAD `Object versionId`-Anforderung.

Example – Abrufen der Metadaten eines versionierten Objekts

Die folgende Anforderung ruft die Metadaten der Version `3HL4kqCxf3vjVBH40N1jfkD` von `my-image.jpg` ab.

```
HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40N1jfkD HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Nachfolgend ist eine Beispielantwort angezeigt.

```
HTTP/1.1 200 OK
x-amz-id-2: ef8yU9AS1ed40pIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXroDTDmjVBH40N1jfkD
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

Wiederherstellen früherer Versionen

Sie können Versioning verwenden, um frühere Versionen eines Objekts abzurufen. Hierfür gibt es zwei Ansätze:

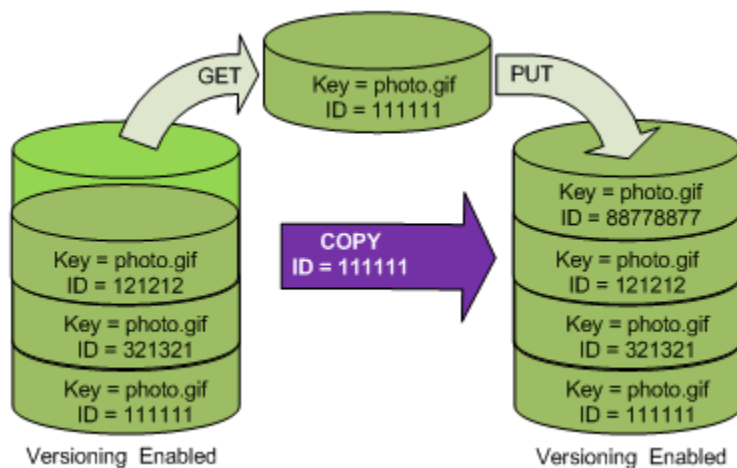
- Kopieren einer vorhergehenden Version des Objekts in denselben Bucket.

Das kopierte Objekt wird zur aktuellen Version dieses Objekts, und alle Objektversionen werden beibehalten.

- Dauerhaftes Löschen der aktuellen Version des Objekts

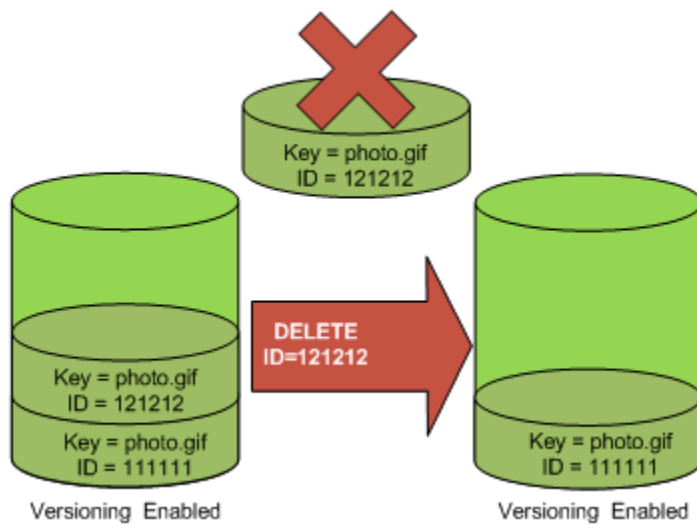
Wenn Sie die aktuelle Objektversion löschen, wandeln Sie letztlich die nicht vorherige Version in die aktuelle Version dieses Objekts um.

Alle Objektversionen werden aufbewahrt, deshalb können Sie eine frühere Version zur aktuellen Version machen, indem Sie eine spezifische Version des Objekts in denselben Bucket kopieren. In der folgenden Abbildung wird das Quellobjekt (ID = 111111) in denselben Bucket kopiert. Amazon S3 stellt eine neue ID (88778877) bereit und diese wird zur aktuellen Version des Objekts. Der Bucket enthält jetzt also die ursprüngliche Objektversion (111111) und kopiert seine Kopie (88778877). Weitere Informationen zum Abrufen einer früheren Version und zum anschließenden Hochladen, um sie zur aktuellen Version zu machen, finden Sie unter [Objektversionen aus einem versionsfähigen Bucket abrufen](#) und [Objekte hochladen](#).



Eine nachfolgende GET ruft Version 88778877 ab.

Die folgende Abbildung zeigt, wie die aktuelle Version (121212) eines Objekts gelöscht wird, sodass die vorhergehende Version (111111) zum aktuellen Objekt wird. Weitere Informationen zum Löschen eines Objekts finden Sie unter [Löschen eines einzelnen Objekts](#).



Ein weiterer GET ruft die Version 111111 ab.

Note

Wenn Sie Objektversionen in Batches wiederherstellen möchten, können Sie [die CopyObject-Operation](#) verwenden. Die CopyObject-Operation kopiert jedes im Manifest angegebene Objekt. Objekte werden jedoch nicht zwingend in derselben Reihenfolge kopiert, in der sie im Manifest erscheinen. Wenn für versionierte Buckets die Beibehaltung der aktuellen/nicht aktuellen Versionsreihenfolge wichtig ist, sollten Sie zuerst alle nicht aktuellen Versionen kopieren. Kopieren Sie dann, nachdem der erste Auftrag abgeschlossen ist, die aktuellen Versionen in einen nachfolgenden Auftrag.

So stellen Sie frühere Objektversionen wieder her

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus.
4. Wählen Sie Versions (Versionen).

Amazon S3 zeigt alle Versionen für das Objekt an.

5. Aktivieren Sie das Kontrollkästchen neben der Version ID (Versions-ID) für die Versionen, die Sie abrufen möchten.
6. Wählen Sie Actions (Aktionen), wählen Sie Download (Herunterladen) und speichern Sie das Objekt.

Objektversionen können auch in der Objektübersicht angezeigt, heruntergeladen und gelöscht werden. Weitere Informationen finden Sie unter [Anzeigen einer Objektübersicht in der Amazon-S3-Konsole](#).

Important

Sie können den Löschvorgang für ein Objekt nur rückgängig machen, wenn seine aktuelle Version gelöscht wurde. Es ist nicht möglich, das Löschen einer vorherigen Version eines Objekts rückgängig zu machen, das gelöscht wurde. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Verwendung der AWS SDKs

Für Informationen zur Verwendung anderer AWS SDKs, siehe [AWS Entwicklerzentrum](#).

Python

Im folgenden Python-Codebeispiel wird die vorherige Version eines versionierten Objekts wiederhergestellt, indem alle Versionen gelöscht werden, die nach der angegebenen Rollback-Version aufgetreten sind.

```
def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
```

```
# Versions must be sorted by last_modified date because delete markers are
# at the end of the list even when they are interspersed in time.
versions = sorted(
    bucket.object_versions.filter(Prefix=object_key),
    key=attrgetter("last_modified"),
    reverse=True,
)

logger.debug(
    "Got versions:\n%s",
    "\n".join(
        [
            f"\t{version.version_id}, last modified {version.last_modified}"
            for version in versions
        ]
    ),
)

if version_id in [ver.version_id for ver in versions]:
    print(f"Rolling back to version {version_id}")
    for version in versions:
        if version.version_id != version_id:
            version.delete()
            print(f"Deleted version {version.version_id}")
        else:
            break

    print(f"Active version is now {bucket.Object(object_key).version_id}")
else:
    raise KeyError(
        f"{version_id} was not found in the list of versions for "
        f"{object_key}."
    )
```

Löschen von Objekten aus einem versioning-fähigen Bucket

Sie können Objektversionen aus Amazon-S3-Buckets löschen, wann immer Sie möchten. Sie können auch Lebenszyklus-Konfigurationsregeln für Objekte definieren, die einen definierten Lebenszyklus haben, um Amazon S3 aufzufordern, die aktuellen Objektversionen ablaufen zu lassen, oder die nicht aktuellen Objektversionen ständig zu entfernen. Wenn in Ihrem Bucket die

Versionsverwaltung aktiviert ist oder die Versionierung ausgesetzt ist, funktionieren die Aktionen zur Lebenszykluskonfiguration wie folgt:

- Die Aktion `Expiration` gilt für die aktuelle Objektversion. Statt die aktuelle Objektversion zu löschen, behält Amazon S3 die aktuelle Version als nicht aktuelle Version bei, indem es eine Löschmarkierung hinzufügt, die anschließend zur aktuellen Version wird.
- Die `NoncurrentVersionExpiration`-Aktion wird nur auf nicht aktuelle Objektversionen angewendet, und Amazon S3 entfernt diese Objektversionen dauerhaft. Dauerhaft entfernte Objekte können nicht wiederhergestellt werden.

Weitere Informationen zu S3 Lifecycle finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#) und [Beispiele der S3-Lebenszyklus-Konfiguration](#).

Um zu sehen, wie viele aktuelle und nicht aktuelle Objektversionen Ihre Buckets enthalten, können Sie die Metriken von Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter [Verwenden von S3 Storage Lens zur Optimierung Ihrer Speicherkosten](#). Eine vollständige Liste der Metriken finden Sie im [Glossar der S3-Storage-Lens-Metriken](#).

Note

Die normalen Amazon S3 S3-Tarife gelten für jede Version eines Objekts, die gespeichert und übertragen wird, einschließlich nicht aktueller Objektversionen. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Löschen von Anwendungsfällen

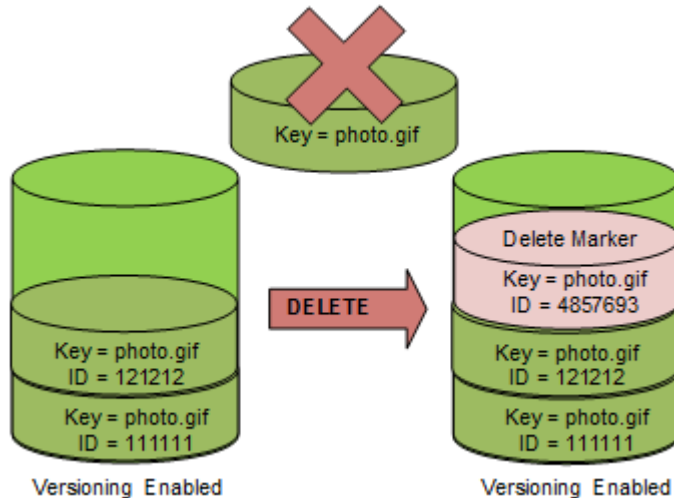
Eine DELETE-Anforderung hat die folgenden Anwendungsfälle:

- Wenn das Versioning aktiviert ist, kann ein einfaches DELETE ein Objekt nicht dauerhaft löschen. (Eine einfache DELETE-Anforderung ist eine Anforderung, die keine Versions-ID angibt.) Stattdessen fügt Amazon S3 eine Löschmarkierung in den Bucket ein, und die Löschmarkierung wird zur aktuellen Objektversion mit einer neuen ID.

Wenn Sie versuchen, GET für ein Objekt auszuführen, dessen aktuelle Version eine Löschmarkierung ist, verhält sich Amazon S3, als wäre das Objekt gelöscht worden (obwohl es

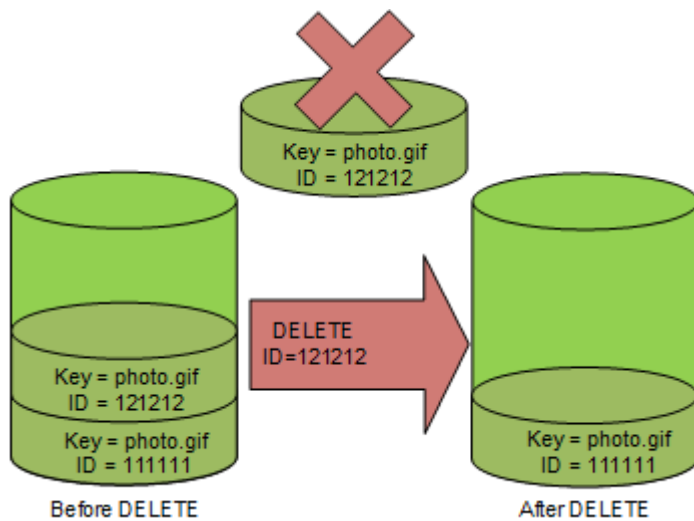
nicht vollständig entfernt wurde), und gibt einen 404-Fehler zurück. Weitere Informationen finden Sie unter [Arbeiten mit Löschmarkierungen](#).

Die folgende Abbildung zeigt, dass ein einfaches DELETE das spezifizierte Objekt nicht wirklich löscht. Stattdessen fügt Amazon S3 eine Löschmarkierung ein.



- Um Objektversionen dauerhaft zu löschen, müssen Sie verwenden `DELETE Object versionId`.

Die folgende Abbildung zeigt, dass das beim Löschen einer angegebenen Objektversion dieses Objekt dauerhaft gelöscht wird.



Löschen von Objektversionen

Sie können Objektversionen in Amazon S3 mithilfe der Konsole löschen, AWS SDKs, der RESTAPI, oder der AWS Command Line Interface.

Verwenden der S3-Konsole

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus.
4. Wählen Sie Versions (Versionen).

Amazon S3 zeigt alle Versionen für das Objekt an.

5. Aktivieren Sie das Kontrollkästchen neben der Version ID (Versions-ID) für die Versionen, die Sie permanent löschen möchten.
6. Wählen Sie Delete (Löschen).
7. Geben Sie in Objekte endgültig löschen? **permanently delete** ein.

Warning

Wenn Sie eine Objektversion endgültig löschen, kann die Aktion nicht rückgängig gemacht werden.

8. Wählen Sie Delete objects (Objekte löschen).

Amazon S3 löscht die Objektversion.

Verwendung der AWS SDKs

Beispiele für das Löschen von Objekten mit dem AWS SDKs für Java, .NET, und PHP, siehe [Löschen von Amazon-S3-Objekten](#). Die Beispiele für das Löschen von Objekten in nicht versioning-fähigen und versioning-fähigen Buckets sind dieselben. Für versioning-fähige Buckets weist Amazon S3 jedoch eine Versionsnummer zu. Andernfalls ist die Versionsnummer null.

Informationen zur Verwendung anderer AWS SDKs, siehe [AWS Entwicklerzentrum](#).

Python

Das folgende Beispiel für Python-Code zeigt das dauerhafte Löschen eines versionierten Objekts, indem alle seine Versionen gelöscht werden.

```
def permanently_delete_object(bucket, object_key):
```

```
"""
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Mit dem REST API

Löschen einer spezifischen Objektversion

- Geben Sie in einem DELETE eine Versions-ID ein.

Example – Löschen einer spezifischen Version

Im folgenden Beispiel wird die Version UI0RUfnd89493jJFJ von `photo.gif` gelöscht.

```
DELETE /photo.gif?versionId=UI0RUfnd89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

Verwendung der AWS CLI

Der folgende Befehl löscht ein Objekt namens `test.txt` aus einem Bucket mit dem Namen `amzn-s3-demo-bucket1`. Zum Entfernen einer bestimmte Version eines Objekts müssen Sie der Bucket-Eigentümer sein und die Versions-ID-Subressource verwenden.

```
aws s3api delete-object --bucket amzn-s3-demo-bucket1 --key test.txt --version-id versionID
```

Weitere Informationen zu finden `delete-object` Sie unter [delete-object](#) in der AWS CLI Befehlsreferenz.

Weitere Informationen zum Löschen von Objektversionen finden Sie in den folgenden Themen:

- [Arbeiten mit Löschmarkierungen](#)
- [Entfernen von Löschmarkierungen, um eine ältere Version aktuell zu machen](#)
- [Löschen eines Objekts aus einem Bucket mit MFA aktivierter Löschfunktion](#)

Arbeiten mit Löschmarkierungen

Eine Löschmarkierung in Amazon S3 ist ein Platzhalter (bzw. eine Markierung) für ein versioniertes Objekt, das in einer einfachen DELETE-Anforderung angegeben wurde. Eine einfache DELETE-Anforderung ist eine Anforderung, die keine Versions-ID angibt. Weil sich das Objekt in einem versioning-fähigen Bucket befindet, wurde das Objekt nicht gelöscht. Die Löschmarkierung bewirkt jedoch, dass sich Amazon S3 verhält, als wäre das Objekt gelöscht worden. Sie können einen Amazon S3 API DELETE S3-Aufruf für eine Löschmarke verwenden. Dazu müssen Sie die DELETE Anfrage mit einem stellen AWS Identity and Access Management (IAM) Benutzer oder Rolle mit den entsprechenden Berechtigungen.

Eine Löschmarkierung hat einen Schlüsselnamen (oder Schlüssel) und eine Versions-ID, so wie jedes andere Objekt. Eine Löschmarkierung unterscheidet sich jedoch wie folgt von anderen Objekten:

- Einer Löschmarkierung sind keine Daten zugeordnet.
- Eine Löschmarkierung ist keinem Wert in der Zugriffskontrollliste (ACL) zugeordnet.
- Wenn Sie eine GET-Anforderung für eine Löschmarkierung ausgeben, ruft die GET-Anforderung nichts ab, da eine Löschmarkierung keine Daten enthält. Insbesondere, wenn in Ihrer GET-Anforderung keine `versionId` angegeben ist, erhalten Sie den Fehler 404 (Nicht gefunden).

Für Löschmarkierungen fällt eine Mindestgebühr für die Speicherung in Amazon S3 an. Der Speicherbedarf einer Löschmarkierung ist gleich der Größe des Schlüsselnamens der Löschmarkierung. Ein Schlüsselname ist eine Folge von Unicode-Zeichen. Die Kodierung UTF -8 für

den Schlüsselnamen fügt Ihrem Bucket für jedes Zeichen im Namen 1 bis 4 Byte Speicherplatz hinzu. Löschkennzeichnungen werden in der S3-Standard-Speicherklasse gespeichert.

Wenn Sie ermitteln möchten, wie viele Löschkennzeichnungen Sie haben und in welcher Speicherklasse diese gespeichert sind, können Sie Amazon S3 Storage Lens verwenden. Weitere Informationen erhalten Sie unter [Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens](#) und [Amazon S3-Storage-Lens-Metrikglossar](#).

Weitere Informationen zu Schlüsselnamen finden Sie unter [Erstellen von Objektschlüsselnamen](#). Informationen zum Löschen von Löschkennzeichnungen finden Sie unter [Verwalten von Löschkennzeichnungen](#).

Nur Amazon S3 kann eine Löschkennzeichnung erstellen. Dies erfolgt immer, wenn Sie eine `DeleteObject`-Anfrage für ein Objekt in einem Bucket mit Versioning oder ausgesetztem Versioning stellen. Das in der `DELETE`-Anforderung angegebene Objekt wird nicht wirklich gelöscht. Stattdessen wird die Löschkennzeichnung zur aktuellen Version des Objekts. Der Schlüsselname (oder Schlüssel) des Objekts wird zum Schlüssel der Löschkennzeichnung.

Wenn Sie ein Objekt abrufen, ohne eine `versionId` in der Anforderung anzugeben, und die aktuelle Version eine Löschkennzeichnung ist, antwortet Amazon S3 mit Folgendem:

- Fehler 404 (Nicht gefunden)
- Ein Antwort-Header `x-amz-delete-marker: true`

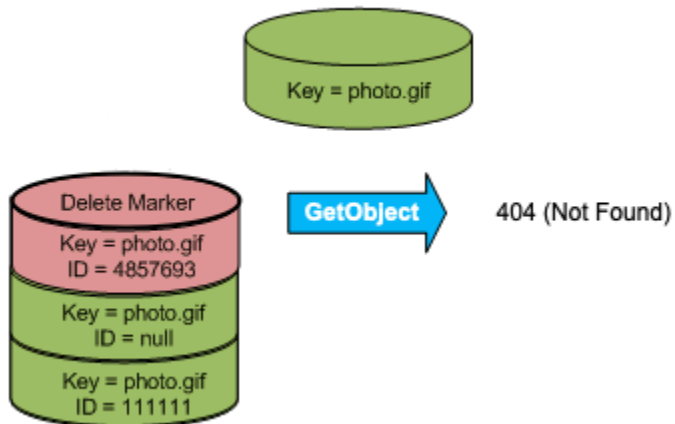
Wenn Sie ein Objekt abrufen, indem Sie eine `versionId` in der Anforderung angeben, und die angegebene Version eine Löschkennzeichnung ist, antwortet Amazon S3 mit Folgendem:

- Ein Fehler 405 (Method Not Allowed)
- Ein Antwort-Header `x-amz-delete-marker: true`
- Ein Antwort-Header `Last-Modified: timestamp` (nur bei Verwendung der [HeadObject](#)-Operationen) [GetObjectAPI](#)

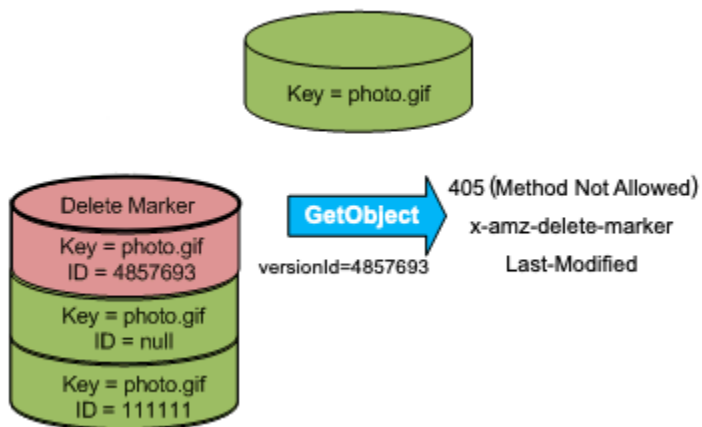
Der Antwort-Header `x-amz-delete-marker: true` teilt Ihnen mit, dass das Objekt, auf das Sie zugegriffen haben, eine Löschkennzeichnung war. Dieser Antwort-Header gibt niemals `false` zurück, denn wenn der Wert `false` lautet, handelt es sich bei der aktuellen oder angegebenen Version des Objekts nicht um eine Löschkennzeichnung.

Der Antwort-Header `Last-Modified` gibt die Erstellungszeit der Löschkennzeichnungen an.

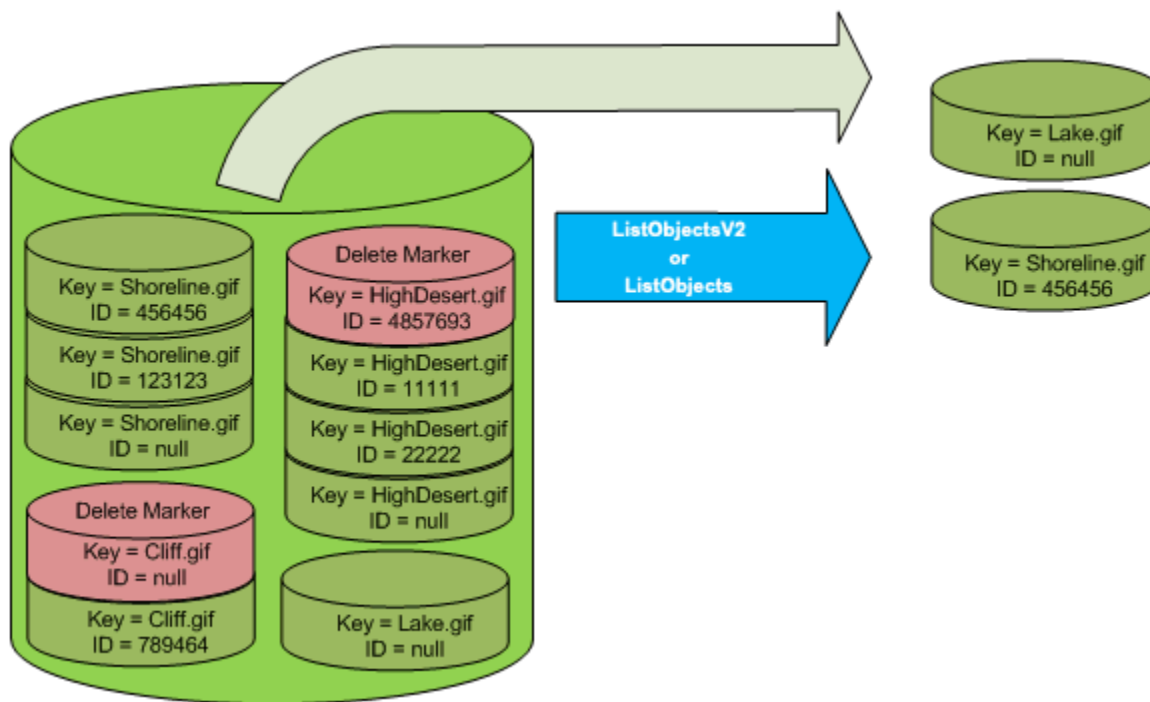
Die folgende Abbildung zeigt, wie ein `GetObject` API Aufruf eines Objekts, dessen aktuelle Version eine Löschmarkierung ist, mit einem 404-Fehler (Not Found) reagiert, und der Antwortheader enthält Folgendes: `x-amz-delete-marker: true`.



Wenn Sie einen `GetObject`-Aufruf für ein Objekt durchführen, indem Sie eine `versionId` in Ihrer Anfrage angeben, und wenn die angegebene Version eine Löschmarkierung ist, antwortet Amazon S3 mit dem Fehler 405 (Methode nicht zulässig) und die Antwort-Header enthalten `x-amz-delete-marker: true` und `Last-Modified: timestamp`.



Die einzige Möglichkeit, Löschmarken (und andere Versionen eines Objekts) aufzulisten, besteht darin, die `versions` Subressource in einem [ListObjectVersions](#) request. Die folgende Abbildung zeigt, dass [ListObjectsV2](#) oder [ListObjects](#) Die Anfrage gibt keine Objekte zurück, deren aktuelle Version eine Löschmarkierung ist.



Verwalten von Löschmarkierungen

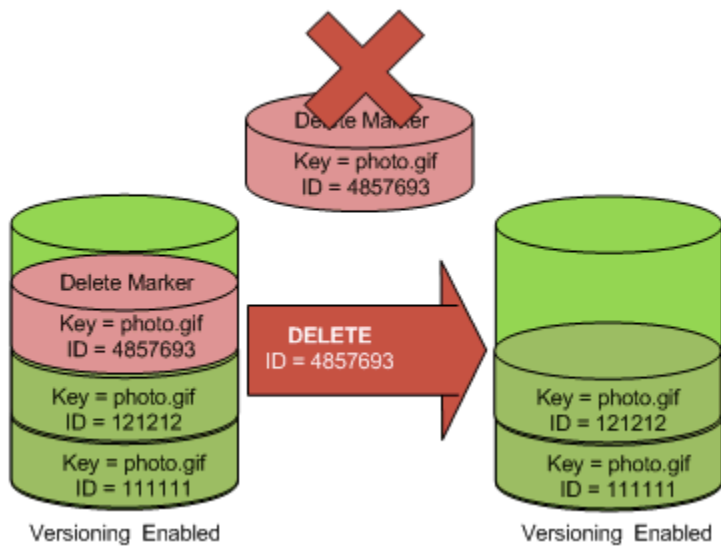
Konfigurieren des Lebenszyklus zum automatischen Bereinigen abgelaufener Löschmarkierungen

Eine abgelaufene Objektlöschmarkierung ist eine, bei der alle Objektversionen gelöscht werden und nur eine einzelne Löschmarkierung erhalten bleibt. Wenn die Lebenszyklus-Konfiguration so eingerichtet ist, dass aktuelle Versionen gelöscht werden, oder die `ExpiredObjectDeleteMarker`-Aktion explizit festgelegt ist, entfernt Amazon S3 die Löschmarkierung des abgelaufenen Objekts. Ein Beispiel finden Sie unter [Beispiel 7: Löschen abgelaufener Löschmarkierungen für Objekte](#).

Entfernen von Löschmarkierungen, um eine ältere Version aktuell zu machen

Wenn Sie ein Objekt in einem für das Versioning geeigneten Bucket löschen, bleiben alle Versionen in dem Bucket und Amazon S3 erstellt eine Löschmarkierung für das Objekt. Um das Löschen des Objekts rückgängig zu machen, müssen Sie diese Löschmarkierung löschen. Weitere Informationen zum Versioning und zu Löschmarkierungen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Um eine Löschmarkierung dauerhaft zu löschen, müssen Sie Ihre Versions-ID in einer `DeleteObject versionId`-Anforderung angeben. Die folgende Abbildung zeigt, dass ein einfaches `DeleteObject versionId` eine Löschmarkierung nicht dauerhaft entfernt.

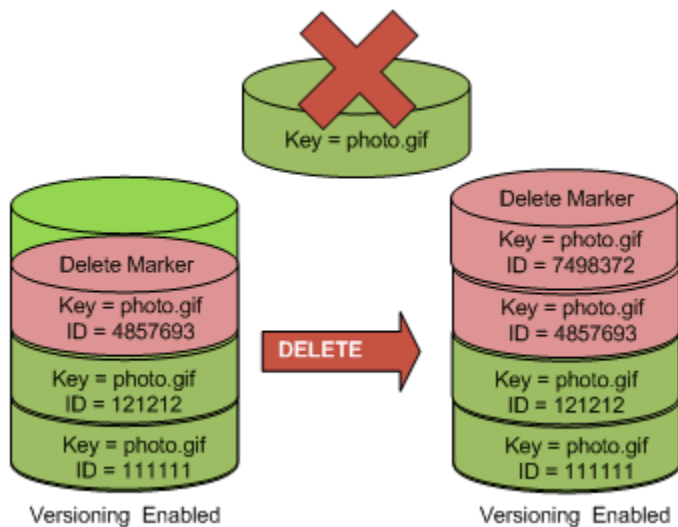


Das Entfernen der Löschmarkierung bewirkt, dass eine einfache GET-Anforderung jetzt die aktuelle Versions-ID (121212) des Objekts abrufen.

Note

Wenn Sie eine `DeleteObject`-Anforderung verwenden, bei der die aktuelle Version eine Löschmarkierung ist (ohne die Versions-ID der Löschmarkierung anzugeben), löscht Amazon S3 die Löschmarkierung nicht, sondern eine andere PUTs Löschmarkierung.

Um eine Löschmarkierung mit einer NULL-Versions-ID zu löschen, müssen Sie das NULL als Versions-ID in der `DeleteObject`-Anforderung übergeben. Die folgende Abbildung zeigt, wie eine einfache `DeleteObject`-Anfrage ohne Versions-ID, bei der die aktuelle Version eine Löschmarkierung ist, nichts entfernt, sondern stattdessen eine zusätzliche Löschmarkierung mit einer eindeutigen Versions-ID (7498372) hinzufügt.



Verwenden der S3-Konsole

Mit den folgenden Schritten können Sie gelöschte Objekte wiederherstellen, die keine Ordner aus Ihrem S3-Bucket sind, einschließlich der Objekte, die sich in diesen Ordnern befinden.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
3. Um eine Liste der Versionen der Objekte in dem Bucket anzuzeigen, wählen Sie den Schalter List versions (Versionen auflisten). Sie sehen die Löschkennzeichnungen für gelöschte Objekte.
4. Um das Löschen eines Objekts rückgängig zu machen, müssen Sie die Löschkennzeichnung löschen. Markieren Sie das Kontrollkästchen neben der Löschkennzeichnung des Objekts, das wiederhergestellt werden soll, und wählen Sie dann Delete (Löschen).
5. Bestätigen Sie den Löschvorgang auf der Seite Delete objects (Objekte löschen) .
 - a. Geben Sie für Permanently delete objects? (Objekte dauerhaft löschen?) **permanently delete** ein.
 - b. Wählen Sie Delete objects (Objekte löschen).

Note

Sie können die Amazon-S3-Konsole nicht verwenden, um das Löschen von Ordnern rückgängig zu machen. Sie müssen das verwenden AWS CLI oder SDK. Beispiele finden Sie

unter [Wie kann ich ein Amazon S3 S3-Objekt abrufen, das in einem Bucket mit aktivierter Versionierung gelöscht wurde?](#) in der AWS Wissenszentrum.

Mit dem REST API

Eine Löschmarkierung dauerhaft entfernen

1. Setzen Sie den Parameter `versionId` auf die ID der Version der Löschmarkierung, die Sie entfernen wollen.
2. Senden Sie eine `DELETE Object versionId`-Anforderung.

Example – Entfernen einer Löschmarkierung

Das folgende Beispiel entfernt die Löschmarkierung für `photo.gif` Version 4857693.

```
DELETE /photo.gif?versionId=4857693 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Wenn Sie eine Löschmarkierung löschen, nimmt Amazon S3 folgendes in die Antwort auf.

```
204 NoContent
x-amz-version-id: versionID
x-amz-delete-marker: true
```

Verwendung der AWS SDKs

Für Informationen zur Verwendung anderer AWS SDKs, siehe [AWS Entwicklerzentrum](#).

Python

Das folgende Python-Codebeispiel zeigt, wie Sie einen Löschmarker von einem Objekt entfernen und damit die neueste nicht-aktuelle Version der aktuellsten Version zur aktuellen Version des Objekts machen.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
```

A versioned object presents as deleted when its latest version is a delete marker.

By removing the delete marker, we make the previous version the latest version and the object then presents as **not** deleted.

Usage is shown in the `usage_demo_single_object` function at the end of this module.

```
:param bucket: The bucket that contains the object.
:param object_key: The object to revive.
"""
# Get the latest version for the object.
response = s3.meta.client.list_object_versions(
    Bucket=bucket.name, Prefix=object_key, MaxKeys=1
)

if "DeleteMarkers" in response:
    latest_version = response["DeleteMarkers"][0]
    if latest_version["IsLatest"]:
        logger.info(
            "Object %s was indeed deleted on %s. Let's revive it.",
            object_key,
            latest_version["LastModified"],
        )
        obj = bucket.Object(object_key)
        obj.Version(latest_version["VersionId"]).delete()
        logger.info(
            "Revived %s, active version is now %s with body '%s'",
            object_key,
            obj.version_id,
            obj.get()["Body"].read(),
        )
    else:
        logger.warning(
            "Delete marker is not the latest version for %s!", object_key
        )
elif "Versions" in response:
    logger.warning("Got an active version for %s, nothing to do.", object_key)
else:
    logger.error("Couldn't get any version info for %s.", object_key)
```

Löschen eines Objekts aus einem Bucket mit MFA aktivierter Löschfunktion

Wenn für die Versionierungskonfiguration eines Buckets die Option MFA Löschen aktiviert ist, muss der Bucket-Besitzer den `x-amz-mfa` Anforderungsheader in Anfragen aufnehmen, um eine Objektversion dauerhaft zu löschen oder den Versionsstatus des Buckets zu ändern. Anfragen, die Folgendes beinhalten, `x-amz-mfa` müssen verwenden. HTTPS

Der Wert des Headers ist die Verkettung der Seriennummer Ihres Authentifizierungsgeräts, eines Leerzeichens und des darauf angezeigten Authentifizierungscode. Wenn sie dies im Anforderungs-Header nicht angeben, schlägt die Anforderung fehl.

Weitere Informationen zu Authentifizierungsgeräten finden Sie unter [Multi-Factor Authentication](#).

Example — Löschen eines Objekts aus einem Bucket mit MFA aktivierter Löschfunktion

Im folgenden Beispiel wird ein Objekt `my-image.jpg` (mit der angegebenen Version) gelöscht, das sich in einem Bucket befindet, für das Löschen aktiviert ist. MFA

Beachten Sie den Abstand zwischen `[SerialNumber]` and `[AuthenticationCode]` Weitere Informationen finden Sie unter [DeleteObject](#) in der Amazon Simple Storage Service API Reference.

```
DELETE /my-image.jpg?versionId=3HL4kqCxf3vjVBH40NrjfkD HTTPS/1.1
Host: bucketName.s3.amazonaws.com
x-amz-mfa: 20899872 301749
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Weitere Informationen zur Aktivierung des MFA Löschvorgangs finden Sie unter [MFALöschen konfigurieren](#).

Konfigurieren von versionierten Objektberechtigungen

Die Berechtigungen für Objekte in Amazon S3 werden auf Versionsebene festgelegt. Jede Version hat ihren eigenen Objekteigentümer. Das Tool AWS-Konto der die Objektversion erstellt, ist der Besitzer. Sie können also unterschiedliche Berechtigungen für unterschiedliche Versionen desselben Objekts einrichten. Dazu geben Sie die Versions-ID des Objekts an, dessen Berechtigungen Sie in einer `PUT Object versionId aCL`-Anforderung setzen wollen. Eine ausführliche Beschreibung und Anweisungen zur Verwendung ACLs finden Sie unter [Identity and Access Management für Amazon S3](#).

Example – Einrichtung von Berechtigungen für eine Objektversion

Die folgende Anforderung legt die Berechtigung für den Empfänger, BucketOwner@amazon.com, auf FULL_CONTROL für den Schlüssel, my-image.jpg, Versions-ID, 3HL4kqtJvjVBH40Nrjfkdfest.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJvjVBH40Nrjfkdf HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>BucketOwner@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Analog dazu müssen Sie, um die Berechtigungen für eine spezifische Objektversion zu erhalten, ihre Versions-ID in einer GET Object versionId acl-Anforderung angeben. Sie müssen diese Versions-ID angeben, weil GET Object acl standardmäßig die Berechtigungen für die aktuelle Version des Objekts zurückgibt.

Example – Abrufen der Berechtigungen für eine bestimmte Objektversion

Im folgenden Beispiel gibt Amazon S3 die Berechtigungen für den Schlüssel, my-image.jpg, Versions-ID, DVBH40Nr8X8gUMLUo zurück.

```
GET /my-image.jpg?versionId=DVBH40Nr8X8gUMLUo&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
```

```
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU
```

Weitere Informationen finden Sie unter [GetObjectAcl](#) in der Amazon Simple Storage Service API Reference.

Arbeiten mit Objekten in einem Bucket mit ausgesetztem Versioning

In Amazon S3 können Sie das Versioning aussetzen, um zu verhindern, dass sich neue Versionen desselben Objekts in einem Bucket ansammeln. Sie könnten dies tun, weil Sie nur eine einzige Version eines Objekts in einem Bucket haben möchten. Oder Sie möchten möglicherweise keine Gebühren für mehrere Versionen anfallen.

Wenn Sie das Versioning aussetzen, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet. Die Themen in diesem Abschnitt erläutern verschiedene Objekt-Vorgänge in einem Bucket mit ausgesetztem Versioning, einschließlich Hinzufügen, Abrufen und Löschen von Objekten.

Weitere Informationen über das S3-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#). Weitere Informationen über das Abrufen von Objektversionen, finden Sie unter [Abrufen von Objektversionen aus einem versioning-fähigen Bucket](#).

Themen

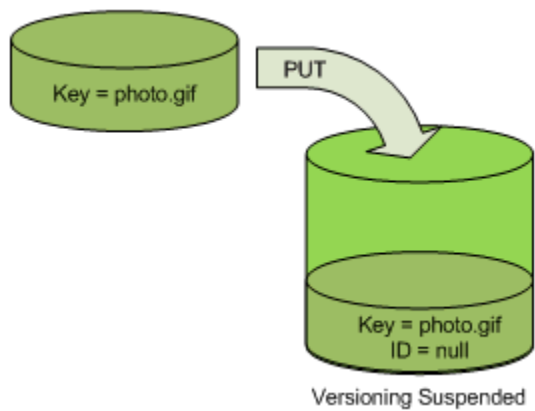
- [Hinzufügen von Objekten zu Buckets mit ausgesetztem Versioning](#)
- [Abrufen von Objekten aus Buckets mit ausgesetztem Versioning](#)
- [Löschen von Objekten aus Buckets mit ausgesetztem Versioning](#)

Hinzufügen von Objekten zu Buckets mit ausgesetztem Versioning

Sie können in Amazon S3 Objekte in Buckets mit ausgesetztem Versioning hinzufügen, um das Objekt mit der Versions-ID null zu erzeugen oder eine Objektversion mit einer übereinstimmenden Versions-ID zu überschreiben.

Nachdem Sie das Versioning für einen Bucket ausgesetzt haben, fügt Amazon S3 automatisch jedem nachfolgend (mit PUT, POST oder CopyObject) in diesem Bucket gespeicherten Objekt die Versions-ID null hinzu.

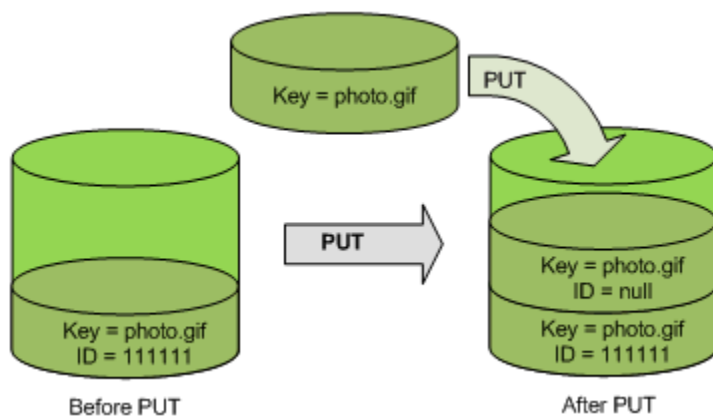
Die folgende Abbildung zeigt, wie Amazon S3 jedem Objekt automatisch die Versions-ID null hinzufügt, wenn es einem Bucket mit ausgesetztem Versioning hinzugefügt wird.



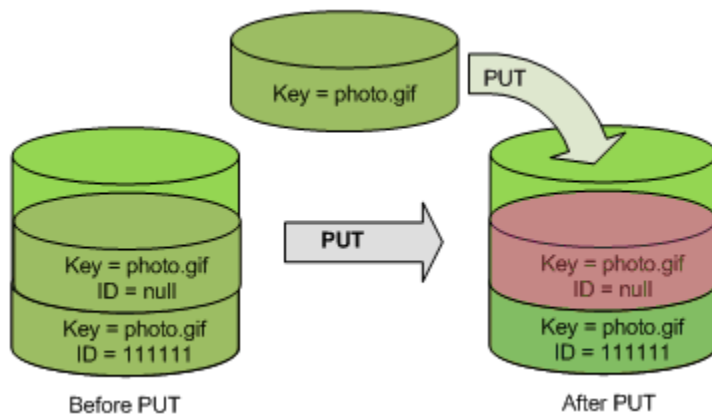
Wenn sich bereits eine Null-Version im Bucket befindet und Sie ein weiteres Objekt mit demselben Schlüssel hinzufügen, überschreibt das hinzugefügte Objekt die ursprüngliche Null-Version.

Wenn es versionsfähige Objekte im Bucket gibt, wird die Version, die Sie mit PUT hinzufügen, zur aktuellen Version des Objekts. Die folgende Abbildung zeigt, wie das Hinzufügen eines Objekts in einen Bucket, der versionsfähige Objekte enthält, das bereits im Bucket enthaltene Objekt nicht überschreibt.

In diesem Fall befand sich Version 111111 bereits im Bucket. Amazon S3 weist dem Objekt, das hinzugefügt und im Bucket gespeichert wird, die Versions-ID null zu. Version 111111 wird nicht überschrieben.



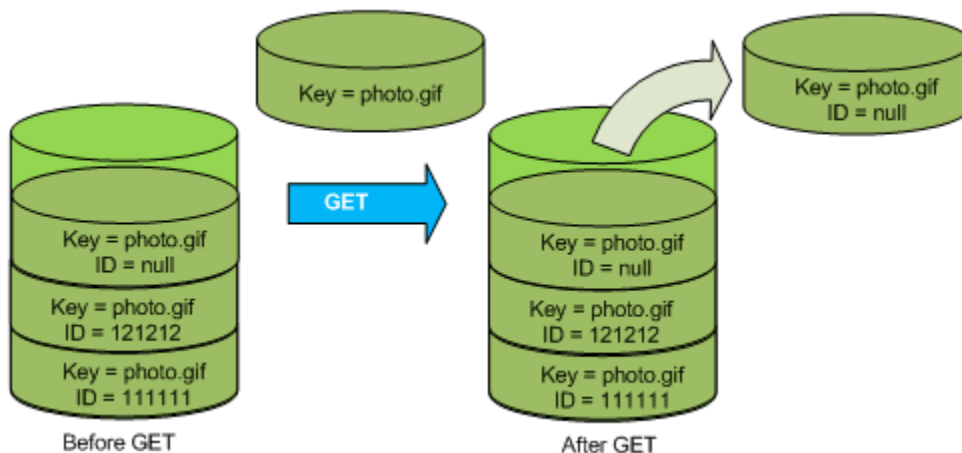
Wenn bereits eine Null-Version in einem Bucket vorhanden ist, wird die Null-Version überschrieben, wie in der folgenden Abbildung gezeigt.



Obwohl der Schlüssel und die Versions-ID (`null`) der Null-Version vor und nach PUT gleich sind, wird der Inhalt der ursprünglich im Bucket gespeicherten Version jedoch durch die Inhalte des Objekts ersetzt, die mit PUT in den Bucket geschrieben wurde.

Abrufen von Objekten aus Buckets mit ausgesetztem Versioning

Eine GET Object-Anforderung gibt die aktuelle Version eines Objekts zurück, unabhängig davon, ob Sie das Versioning für einen Bucket aktiviert haben oder nicht. Die folgende Abbildung zeigt, wie ein einfaches GET die aktuelle Version des Objekts zurückgibt.



Löschen von Objekten aus Buckets mit ausgesetztem Versioning

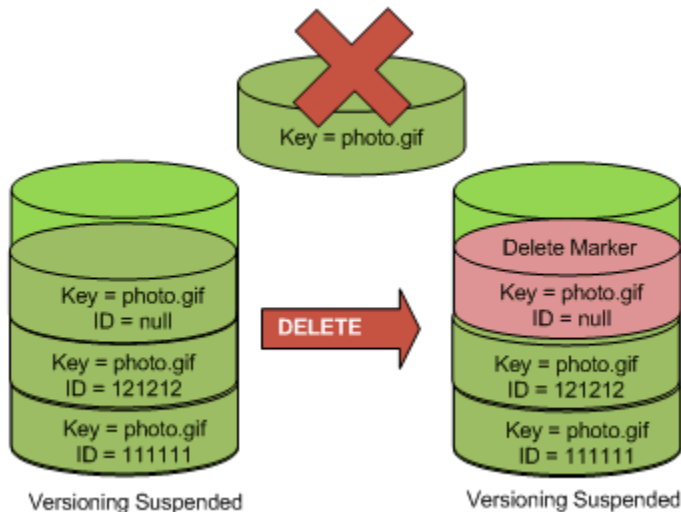
Sie können Objekte aus Buckets mit ausgesetztem Versioning löschen, um ein Objekt mit einer Versions-ID von `null` zu entfernen.

Wenn das Versioning für ein Bucket ausgesetzt ist, gilt für eine DELETE-Anforderung:

- Kann nur ein Objekt entfernen, dessen Versions-ID `null` ist.
- Entfernt nichts, wenn es keine Nullversion des Objekts im Bucket gibt.

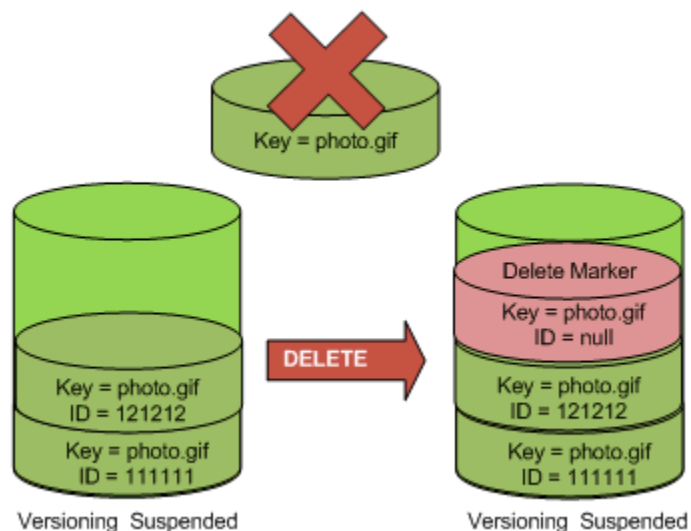
- Fügt eine Löschmarkierung in den Bucket ein.

Die folgende Abbildung zeigt, wie ein einfaches DELETE eine Null-Version entfernt. (Eine einfache DELETE-Anforderung ist eine Anforderung, die keine Versions-ID angibt.) Amazon S3 fügt an seiner Stelle eine Löschmarkierung mit einer Versions-ID von null ein.



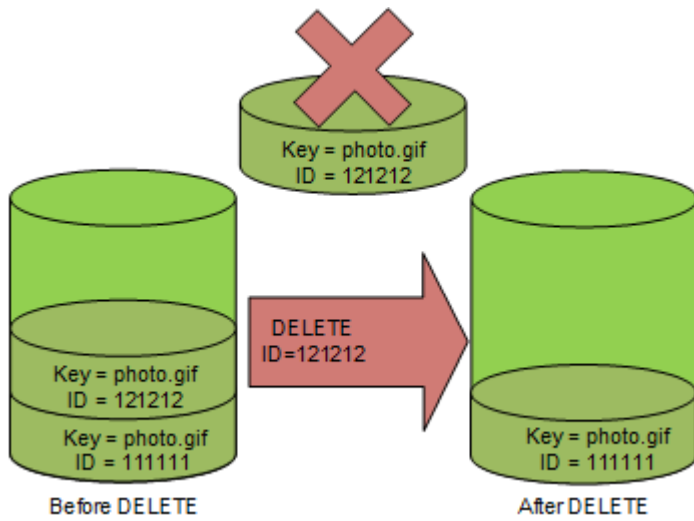
Beachten Sie, dass eine Löschmarkierung keinen Inhalt hat, Sie verlieren also den Inhalt der Nullversion, wenn sie durch eine Löschmarkierung ersetzt wird.

Die folgende Abbildung zeigt einen Bucket, der keine Nullversion enthält. In diesem Fall entfernt DELETE nichts, sondern Amazon S3 fügt einfach eine Löschmarkierung ein.



Selbst in einem Bucket mit ausgesetztem Versioning kann der Bucket-Eigentümer eine spezifische Version dauerhaft löschen, indem er die Versions-ID in der DELETE-Anfrage mit einbezieht. Die

folgende Abbildung zeigt, dass das beim Löschen einer angegebenen Objektversion diese Version des Objekts dauerhaft gelöscht wird. Nur der Bucket-Eigentümer kann eine spezifische Objektversion löschen.



Verwenden von AWS Backup für Amazon S3

Amazon S3 ist nativ integriert in AWS Backup, ein vollständig verwalteter, richtlinienbasierter Service, mit dem Sie Backup-Richtlinien zum Schutz Ihrer Amazon-S3-Daten zentral definieren können. Nachdem Sie Ihre Backup-Richtlinien definiert und ihnen Amazon-S3-Ressourcen zugewiesen haben, automatisiert AWS Backup die Erstellung von Amazon-S3-Backups und speichert sie sicher in einem verschlüsselten Sicherungstresor, den Sie in Ihrem Backup-Plan festlegen.

Sie können die folgenden Aktionen mit AWS Backup für Amazon S3 ausführen:

- Erstellen kontinuierlicher Sicherungen und regelmäßiger Sicherungen. Kontinuierliche Sicherungen sind nützlich für die Point-in-Time Wiederherstellung, und regelmäßige Backups eignen sich, um Ihre Bedürfnisse für die langfristige Datenaufbewahrung zu erfüllen.
- Automatisieren der Planung und Aufbewahrung von Backups durch die zentrale Konfigurierung von Backuprichtlinien.
- Wiederherstellen von Backups von Amazon-S3-Daten zu einem bestimmten Zeitpunkt.

Zusammen mit AWS Backup können Sie die S3-Versionsverwaltung und -Replikation verwenden, um versehentliche Löschungen wiederherzustellen und Ihre eigenen Selbst-Wiederherstellungsvorgänge durchzuführen.

Voraussetzungen

Sie müssen die [S3-Versionsverwaltung](#) in Ihrem Bucket aktivieren, bevor AWS Backup ein Backup erstellen kann.

Note

Wir empfehlen, dass Sie [eine andere Lebenszyklusablaufregel für versionsfähige Buckets](#) festlegen, von denen ein Backup erstellt wird. Wenn Sie keinen Lebenszyklusablaufzeitraum festlegen, können Ihre Amazon-S3-Speicherkosten steigen, weil AWS Backup alle Versionen Ihrer Amazon-S3-Daten beibehält.

Erste Schritte

Informationen zu den ersten Schritten mit AWS Backup für Amazon S3 finden Sie unter [Erstellen von Amazon-S3-Backups](#) im Entwicklerhandbuch für AWS Backup.

Beschränkungen und Einschränkungen

Weitere Informationen über die Einschränkungen finden Sie unter [Erstellen von Amazon-S3-Backups](#) im Entwicklerhandbuch für AWS Backup.

Arbeiten mit archivierten Objekten

Sie können Objekte, auf die selten zugegriffen wird, archivieren, um Ihre Speicherkosten dafür zu senken. Wenn Sie ein Objekt archivieren, wird es in einen kostengünstigen Speicher verschoben. Sie können also nicht in Echtzeit darauf zugreifen.

Auf archivierte Objekte kann zwar nicht in Echtzeit zugegriffen werden, Sie können sie jedoch je nach Speicherklasse innerhalb von Minuten oder Stunden wiederherstellen. Sie können ein archiviertes Objekt mithilfe der Amazon S3 S3-Konsole, S3 Batch Operations RESTAPI, the AWS SDKs, the und AWS Command Line Interface (AWS CLI) wiederherstellen. Anweisungen finden Sie unter [Wiederherstellen eines archivierten Objekts](#).

Amazon-S3-Objekte in den folgenden Speicherklassen oder -stufen werden archiviert. Es kann nicht in Echtzeit darauf zugegriffen werden:

- Die Speicherklasse S3 Glacier Flexible Retrieval
- Die Speicherklasse S3 Glacier Deep Archive

- Die Zugriffsebene S3 Intelligent-Tiering Archive
- Die S3-Intelligent-Tiering-Deep-Archive-Zugriffsebene

Um archivierte Objekte wiederherzustellen, müssen Sie die folgenden Schritte ausführen:

- Für Objekte in den Speicherklassen S3 Glacier Flexible Retrieval und S3-Glacier Deep Archive müssen Sie eine Wiederherstellungsanforderung einleiten und warten, bis eine temporäre Kopie des Objekts verfügbar ist. Wenn eine temporäre Kopie des wiederhergestellten Objekts erstellt wird, bleibt die Speicherkategorie des Objekts unverändert. (Eine [HeadObjectGetObject](#) API-Operationsanforderung gibt S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive als Speicherkategorie zurück.)
- Für Objekte in den Zugriffsebenen S3 Intelligent-Tiering Archive und S3 Intelligent-Tiering Deep Archive müssen Sie eine Wiederherstellungsanforderung einleiten und warten, bis das Objekt in die Stufe für häufigen Zugriff verschoben wird.

Weitere Informationen darüber, wie sich alle Amazon S3-Speicherklassen vergleichen, finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#). Weitere Informationen zu S3 Intelligent-Tiering finden Sie unter [the section called "So funktioniert S3 Intelligent-Tiering"](#).

Wiederherstellen von Objekten aus S3 Glacier

Wenn Sie S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive verwenden, stellt Amazon S3 eine temporäre Kopie des Objekts nur für die angegebene Dauer wieder her. Anschließend wird die Kopie des wiederhergestellten Objekts gelöscht. Sie können den Ablaufzeitraum einer wiederhergestellten Kopie durch die erneute Ausgabe einer Wiederherstellungsanforderung ändern. In diesem Fall aktualisiert Amazon S3 den Ablaufzeitraum relativ zum aktuellen Zeitraum.

Note

Bei einer Wiederherstellung eines archivierten Objekts von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive zahlen Sie sowohl für das archivierte Objekt als auch für die temporär wiederhergestellte Kopie. Informationen zu Preisen finden Sie unter [Amazon S3 – Preise](#).

Wiederherstellen von Objekten aus S3 Intelligent-Tiering

Wenn Sie ein Objekt aus der S3 Intelligent-Tiering Archive Access Tier oder S3 Intelligent-Tiering Deep Archive Access Tier wiederherstellen, wird das Objekt zurück in die S3 Intelligent-Tiering Frequent Access Tier verschoben. Wenn dnach 30 aufeinanderfolgenden Tagen nicht auf das Objekt zugegriffen wird, wird es automatisch in Infrequent Access Tier verschoben. Das Objekt wechselt nach mindestens 90 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die S3 Intelligent-Tiering Archive Access Tier. Wenn mindestens 180 aufeinanderfolgende Tage nicht auf ein Objekt zugegriffen wird, wird das Objekt in Deep Archive Access Tier verschoben.

Note

Im Gegensatz zu den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive akzeptieren Wiederherstellungsanforderungen für S3-Intelligent-Tiering-Objekte den Days-Wert nicht.

Verwenden von S3-Batch-Operationen mit Wiederherstellungsanforderungen

Zur Wiederherstellung von mehr als einem Amazon S3-Objekt in einer einzigen Anforderung können Sie S3-Batch-Vorgänge nutzen. Sie stellen S3-Batch-Vorgänge eine Liste von Objekten zur Verfügung, für die Vorgänge ausgeführt werden sollen. S3 Batch Operations ruft die entsprechende API Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Wiederherstellungszeit

Amazon S3 berechnet die Ablaufzeit der wiederhergestellten Objektkopie durch Addition der Anzahl der in der Wiederherstellungsanforderung angegebenen Tage zum Zeitpunkt, wenn die angeforderte Wiederherstellung abgeschlossen wird. Amazon S3 rundet dann die sich ergebende Zeit auf den nächsten Tag um Mitternacht koordinierte Weltzeit (UTC) ab. Nehmen wir zum Beispiel an, dass am 15. Oktober 2012 um 10:30 Uhr UTC eine wiederhergestellte Objektkopie erstellt wurde und der Wiederherstellungszeitraum auf 3 Tage festgelegt wurde. In diesem Fall läuft die wiederhergestellte Kopie am 19. Oktober 2012 um 00:00 Uhr UTC ab. Zu diesem Zeitpunkt löscht Amazon S3 die Objektkopie.

Die Zeit, die für den Abschluss eines Wiederherstellungsauftrags benötigt wird, hängt davon ab, welche Archivspeicherklasse oder Speicherebene Sie verwenden und welche Abrufoption Sie angeben: Beschleunigt (nur für S3 Glacier Flexible Retrieval und S3 Intelligent Tiering Archive Access verfügbar), Standard, oder Bulk. Weitere Informationen finden Sie unter [Archiv-Abrufoptionen](#).

Sie können über Amazon-S3-Ereignisbenachrichtigungen über den Abschluss der Wiederherstellung benachrichtigt werden. Weitere Informationen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Themen

- [Archiv-Abrufoptionen](#)
- [Wiederherstellen eines archivierten Objekts](#)

Archiv-Abrufoptionen

Nachfolgend finden Sie die verfügbaren Abrufoptionen bei der Wiederherstellung eines archivierten Objekts in Amazon S3:

- **Beschleunigt:** Greifen Sie schnell auf Ihre Daten zu, die in der Speicherklasse „S3 Glacier Flexible Retrieval“ oder der S3 Intelligent-Tiering-Zugriffsebene „Archive“ gespeichert sind. Sie können diese Option verwenden, wenn gelegentliche dringende Anforderungen für eine Teilmenge von Archiven erforderlich sind. Daten, die unter Verwendung von beschleunigten Abrufen abgerufen werden, stehen normalerweise innerhalb von 1 bis 5 Minuten zur Verfügung, außer es handelt sich um die größten archivierten Objekte (250 MB+).

Note

Beschleunigte Abrufe sind eine Premium-Funktion und werden zum Tarif für beschleunigte Anfragen und Abrufe berechnet.

Informationen zu Amazon-S3-Preisen finden Sie unter [Amazon S3-Preise](#).

Die bereitgestellte Kapazität hilft sicherzustellen, dass für Expedited-Abrufe aus S3 Glacier Flexible Retrieval Abrufkapazität verfügbar ist, wenn Sie sie benötigen. Weitere Informationen finden Sie unter [Bereitgestellte Kapazität](#).

- **Standard:** Greifen Sie auf Ihre archivierten Objekte innerhalb einiger Stunden zu. Standard ist die Standardoption für Abrufanforderungen, in denen keine Abrufoption angegeben ist. Die

Standardabrufe enden in der Regel innerhalb von 3 bis 5 Stunden für Objekte, die in der S3 Glacier Flexible Retrieval Speicherklasse oder der S3 Intelligent Tiering Archive Access Stufe gespeichert sind. Diese Abrufe enden in der Regel innerhalb von 12 Stunden für Objekte, die in der Speicherklasse S3 Glacier Deep Archive oder S3 Intelligent-Tiering Deep Archive Access Tier gespeichert sind. Standardabrufe sind für Objekte, die in S3 Intelligent-Tiering gespeichert sind, kostenlos.

Note

- Bei Objekten, die in der Speicherklasse S3 Glacier Flexible Retrieval oder der Stufe S3 Intelligent-Tiering Archive Access gespeichert sind, beginnen Standardabrufe, die mithilfe des Wiederherstellungsvorgangs S3 Batch Operations initiiert wurden, in der Regel innerhalb von Minuten und enden innerhalb von 3 bis 5 Stunden.
- Bei Objekten der S3 Glacier Deep Archive Archive-Speicherklasse oder der S3 Intelligent-Tiering Deep Archive Access-Stufe beginnen Standardabrufe, die mithilfe des Batch Operations-Wiederherstellungsvorgangs initiiert wurden, in der Regel innerhalb von 9 Stunden und enden innerhalb von 12 Stunden.

- **Bulk:** Greifen Sie über die kostengünstigste Abrufoption von Amazon S3 Glacier auf Ihre Daten zu. Mit Massenabrufen können Sie Daten in großen Mengen (sogar im Petabyte-Bereich) kostengünstig abrufen.

Bei Objekten, die in der Speicherklasse S3 Glacier Flexible Retrieval oder der Stufe S3 Intelligent-Tiering Archive Access gespeichert sind, werden Massenabrufe in der Regel innerhalb von 5–12 Stunden abgeschlossen. Bei Objekten, die in der Speicherklasse S3 Glacier Deep Archive oder der S3 Intelligent-Tiering Deep Archive Access-Stufe gespeichert sind, werden diese Abrufe in der Regel innerhalb von 48 Stunden abgeschlossen.

Massenabrufe sind für Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Intelligent-Tiering gespeichert sind, kostenlos.

Die folgende Tabelle fasst die Archivabrufoptionen zusammen. Informationen zu Preisen finden Sie unter [Amazon S3 – Preise](#).

Um einen oder Bulk -Abruf Standard durchzuführen Expedited, setzen Sie das Tier Anforderungselement in der [RestoreObject](#) REST API Operationsanforderung auf die gewünschte Option oder auf die entsprechende Option in () oder. AWS Command Line Interface AWS CLI AWS

SDKs Wenn Sie bereitgestellte Durchsatzkapazität gekauft haben, werden alle Expedited-Abrufe automatisch über Ihre bereitgestellte Kapazität erledigt.

Bereitgestellte Kapazität

Die bereitgestellte Kapazität hilft sicherzustellen, dass für Ihre Expedited-Abrufe aus S3 Glacier Flexible Retrieval Abrufkapazität verfügbar ist, wenn Sie sie benötigen. Jede Kapazitätseinheit sorgt dafür, dass mindestens drei beschleunigte Abrufe alle 5 Minuten ausgeführt werden können, und sie bietet einen Abrufdurchsatz von bis zu 150 Megabyte pro Sekunde (). MBps

Denken Sie über den Kauf bereitgestellter Abrufkapazität nach, wenn Ihr Workload einen sehr zuverlässigen und vorhersehbaren Zugriff auf eine Untermenge Ihrer Daten innerhalb von Minuten erforderlich macht. Ohne bereitgestellte Kapazität werden beschleunigte Abrufe in Zeiten hoher Nachfrage möglicherweise nicht akzeptiert. Wenn Sie unbedingt Zugriff auf beschleunigte Abrufe benötigen, sollten Sie eine bereitgestellte Abrufkapazität kaufen.

Bereitgestellte Kapazitätseinheiten werden einem zugewiesen. AWS-Konto Daher sollte der Antragsteller des beschleunigten Datenabrufs die bereitgestellte Kapazitätseinheit erwerben, nicht der Bucket-Eigentümer.

Sie können bereitgestellte Kapazität erwerben, indem Sie die Amazon S3-Konsole, die Amazon S3 Glacier-Konsole, den REST API Vorgang [Purchase Provisioned Capacity](#) AWS SDKs, den oder den verwenden. AWS CLI Weitere Informationen zu den Preisen für die bereitgestellte Kapazität finden Sie in der [Amazon S3-Preisliste](#).

Geschwindigkeiten für Wiederherstellungsanforderungen für S3 Glacier

Wenn Sie Wiederherstellungsanforderungen für Objekte initiieren, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert sind, wird ein Kontingent für Wiederherstellungsanforderungen für Ihr AWS-Konto angewendet. S3 Glacier unterstützt Wiederherstellungsanforderungen mit einer Geschwindigkeit von 1 000 Transaktionen pro Sekunde. Wenn diese Rate überschritten wird, werden andernfalls gültige Anfragen gedrosselt oder zurückgewiesen und Amazon S3 gibt einen `ThrottlingException`-Fehler zurück.

Optional können Sie auch S3-Massenoperationen verwenden, um eine große Anzahl von Objekten, die in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert sind, mit einer einzigen Anforderung abzurufen. Weitere Informationen finden Sie unter [Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch](#).

Wiederherstellen eines archivierten Objekts

Amazon-S3-Objekte in den folgenden Speicherklassen oder -stufen werden archiviert. Es kann nicht in Echtzeit darauf zugegriffen werden:

- Die Speicherklasse S3 Glacier Flexible Retrieval
- Die Speicherklasse S3 Glacier Deep Archive
- Die Zugriffsebene S3 Intelligent-Tiering Archive
- Die S3-Intelligent-Tiering-Deep-Archive-Zugriffsebene

Ein Zugriff auf in der Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeicherte Amazon-S3-Objekte ist nicht unmittelbar möglich. Um auf ein Objekt in diesen Speicherklassen zuzugreifen, müssen Sie eine temporäre Kopie des Objekts für eine angegebene Dauer (Anzahl von Tagen) in seinem S3-Bucket wiederherstellen. Wenn Sie eine permanente Kopie des Objekts benötigen, stellen Sie das Objekt wieder her und legen Sie dann eine Kopie davon in Ihrem Amazon S3-Bucket an. Das Kopieren wiederhergestellter Objekte wird in der Amazon-S3-Konsole nicht unterstützt. Verwenden Sie für diese Art von Kopiervorgang AWS Command Line Interface (AWS CLI) AWS SDKs, das oder das. REST API Wenn Sie keine Kopie erstellen und die Speicherklasse ändern, wird das Objekt weiterhin in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert. Informationen zur Verwendung dieser Speicherklassen finden Sie unter [Speicherklassen für Objekte, auf die selten zugegriffen wird](#).

Für Zugriff auf Objekte in den S3 Intelligent-Tiering Archive Access und Deep Archive Access müssen Sie eine Wiederherstellungsanforderung initiieren und warten, bis das Objekt in die Stufe für häufigen Zugriff verschoben wird, um darauf zuzugreifen. Wenn Sie ein Objekt aus den Stufen Archive Access oder Deep Archive Access wiederherstellen, wird das Objekt zurück in die Stufe für häufige Zugriffe übergehen. Informationen zur Verwendung dieser Speicherklassen finden Sie unter [Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekanntem Zugriffsmustern](#).

Allgemeine Informationen über archivierte Objekte finden Sie unter [Arbeiten mit archivierten Objekten](#).

 Note

- Wenn Sie ein archiviertes Objekt aus den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, zahlen Sie sowohl für das archivierte Objekt als auch für die Kopie, die Sie vorübergehend wiederhergestellt haben.
- Wenn Sie ein Objekt aus S3 Intelligent-Tiering wiederherstellen, fallen keine Abrufgebühren für Standard- oder Massenabrufe an.
- Nachfolgende Wiederherstellungsanforderungen, die für archivierte Objekte aufgerufen werden, die bereits wiederhergestellt wurden, werden als Anfragen abgerechnet. GET Informationen zu Preisen finden Sie unter [Amazon S3 – Preise](#).

Wiederherstellen eines archivierten Objekts

Sie können ein archiviertes Objekt mithilfe der Amazon S3-Konsole, Amazon S3 RESTAPI, the AWS SDKs, the AWS Command Line Interface (AWS CLI) oder S3 Batch Operations wiederherstellen.

Verwenden der S3-Konsole


Wiederherstellung von Objekten mithilfe der Amazon-S3-Konsole

Gehen Sie wie folgt vor, um ein Objekt wiederherzustellen, das in den Speicherklassen „S3 Glacier Flexible Retrieval“ oder „S3 Glacier Deep Archive“ bzw. in den Speicherstufen „S3 Intelligent-Tiering Archive Access“ oder „Deep Archive Access“ archiviert wurde.

So stellen Sie ein archiviertes Objekt wieder her

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie wiederherstellen möchten.
4. Wählen Sie in der Liste Object (Objekt) das/die Objekt(e) aus, das/die Sie wiederherstellen möchten. Wählen Sie anschließend Actions (Aktionen) und dann Initiate restore (Wiederherstellung initiieren) aus.

5. Wenn Sie von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, geben Sie im Dialogfeld Anzahl der Tage, die die wiederhergestellte Kopie verfügbar ist die Anzahl der Tage ein, die die archivierten Daten verfügbar sein sollen.
6. Führen Sie bei Abrufoptionen einen der folgenden Schritte aus:
 - Wählen Sie Massenabruf oder Standardabruf und dann Wiederherstellen starten aus.
 - Wählen Sie Expedited retrieval (Expedited-Abruf) (nur für S3 Glacier Flexible Retrieval oder S3 Intelligent Tiering Archive Access verfügbar). Wenn Sie ein Objekt in S3 Glacier Flexible Retrieval wiederherstellen, können Sie auswählen, ob Sie bereitgestellte Kapazität für Ihren Expressabruf erwerben möchten. Wenn Sie bereitgestellte Kapazität erwerben möchten, fahren Sie mit dem nächsten Schritt fort. Wenn Sie dies nicht wünschen, wählen Sie Wiederherstellen starten aus.

 Note

Objekte aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access werden automatisch in der Stufe für häufige Zugriffe wiederhergestellt.

7. (Optional) Wenn Sie ein Objekt in S3 Glacier Flexible Retrieval wiederherstellen und Beschleunigter Abruf auswählen, können Sie sich entscheiden, ob Sie bereitgestellte Kapazität erwerben möchten. Die bereitgestellte Kapazität steht nur für Objekte in S3 Glacier Flexible Retrieval zur Verfügung. Wenn Sie bereits über bereitgestellte Kapazität verfügen, wählen Sie Wiederherstellen starten aus, um einen bereitgestellten Abruf zu starten.

Wenn Sie bereitgestellte Durchsatzkapazität gekauft haben, werden alle Ihre Expedited-Abrufe automatisch über Ihre bereitgestellte Kapazität erledigt. Weitere Informationen finden Sie unter [Bereitgestellte Kapazität](#).

- Wenn Sie nicht über bereitgestellte Kapazität verfügen und auch keine kaufen möchten, wählen Sie Wiederherstellen starten aus.
- Wenn Sie keine bereitgestellte Kapazität haben, aber bereitgestellte Kapazitätseinheiten (PCUs) kaufen möchten, wählen Sie Kaufen. PCUs Wählen Sie im PCUs Dialogfeld Kaufen aus, wie viele PCUs Sie kaufen möchten, bestätigen Sie Ihren Kauf und wählen Sie dann Kaufen aus. PCUs Wenn Sie die Meldung Kauf erfolgreich erhalten, wählen Sie Wiederherstellen starten aus, um den bereitgestellten Abruf zu starten.

Verwenden Sie den AWS CLI

Objekte aus S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen

Im folgenden Beispiel wird der `restore-object` Befehl verwendet, um das Objekt wiederherzustellen *dir1/example.obj* 25 Tage im Eimer. *amzn-s3-demo-bucket*

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{"Days":25,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Wenn die im Beispiel verwendete JSON Syntax zu einem Fehler auf einem Windows-Client führt, ersetzen Sie die Wiederherstellungsanforderung durch die folgende Syntax:

```
--restore-request Days=25,GlacierJobParameters={"Tier"="Standard"}
```

Objekte aus S3 Intelligent-Tiering Archive Access und Deep Archive Access wiederherstellen

Im folgenden Beispiel wird der `restore-object` Befehl verwendet, um das Objekt wiederherzustellen *dir1/example.obj* im Bucket zur Stufe für häufigen *amzn-s3-demo-bucket* Zugriff.

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{}'
```

Note

Im Gegensatz zu den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive akzeptieren Wiederherstellungsanforderungen für S3-Intelligent-Tiering-Objekte den Days-Wert nicht.

Den Wiederherstellungsstatus überwachen

Sie können den folgenden `head-object`-Befehl verwenden, um den Status Ihrer `restore-object`-Anforderung zu überwachen:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Weitere Informationen finden Sie unter [restore-object](#) in der Referenz zum AWS CLI -Befehl.

Mit dem REST API

Amazon S3 bietet Ihnen einen API Vorgang, mit dem Sie die Wiederherstellung eines archivierten Objekts initiieren können. Weitere Informationen finden Sie [RestoreObject](#) in der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Beispiele für die Wiederherstellung archivierter Objekte in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive mit dem AWS SDKs finden Sie unter [Verwenden Sie RestoreObject mit einem AWS SDKoder CLI](#).

Verwendung von S3 Batch Operations

Zur Wiederherstellung von mehr als einem archivierten Objekt in einer einzigen Anforderung können Sie S3 Batch Operations nutzen. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, für die Vorgänge ausgeführt werden sollen. S3 Batch Operations ruft die entsprechende API Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Um einen Batch-Operations-Auftrag zu erstellen, benötigen Sie ein Manifest, das nur die Objekte enthält, die Sie wiederherstellen möchten. Sie können mithilfe von S3 Inventory ein Manifest erstellen oder eine CSV Datei mit den erforderlichen Informationen bereitstellen. Weitere Informationen finden Sie unter [the section called "Angeben eines Manifests"](#).

Bevor Sie S3-Batch-Operations-Aufträge erstellen und ausführen, müssen Sie Amazon S3 die Erlaubnis erteilen, S3-Batch-Operations in Ihrem Namen durchzuführen. Die erforderlichen Berechtigungen finden Sie unter [the section called "Gewähren von Berechtigungen"](#).

Note

Batch-Operations-Aufträge können entweder mit Objekte der Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive oder mit Objekten der Speicherstufen S3 Intelligent-Tiering Archive Access und Deep Archive Access arbeiten. Batch Operations kann im selben Auftrag nicht für beide Typen von archivierten Objekten ausgeführt werden. Um Objekte beider Typen wiederherzustellen, müssen Sie separate Batchoperations-Aufgaben erstellen.

Weitere Informationen zur Verwendung von Batch Operations zum Wiederherstellen archivierter Objekte finden Sie unter [the section called "Wiederherstellen von Objekten"](#).

So erstellen Sie einen S3-Auftrag „Initiate Restore Object Batch Operations“

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie Job erstellen aus.
4. Wählen Sie für AWS-Region die Region aus, in der Sie Ihren Auftrag erstellen möchten.
5. Wählen Sie unter Manifestformat das zu verwendende Manifest aus.
 - Wenn Sie den S3-Inventarbericht wählen, geben Sie den Pfad zu dem `manifest.json` Objekt ein, das Amazon S3 als Teil des CSV -formatierten Inventarberichts generiert hat. Wenn Sie nicht das aktuelle Manifest verwenden möchten, können Sie optional die Versions-ID des `manifest.json`-Objekts angeben.
 - Wenn Sie möchten CSV, geben Sie den Pfad zu einem Manifest-Objekt im CSV -Format ein. Das Manifestobjekt muss das in der Konsole beschriebene Format befolgen. Wenn Sie nicht die aktuelle Version verwenden möchten, können Sie optional die Versions-ID des Manifestobjekts angeben.
6. Wählen Sie Weiter aus.
7. Wählen Sie im Abschnitt Operation die Option Wiederherstellen aus.
8. Wählen Sie im Abschnitt Wiederherstellen bei Quelle wiederherstellen entweder Glacier Flexible Retrieval oder Glacier Deep Archive bzw. die Stufe Intelligent-Tiering Archive Access oder Deep Archive Access aus.

Wenn Sie Glacier Flexible Retrieval oder Glacier Deep Archive auswählen, geben Sie eine Zahl für die Anzahl der Tage, an denen die wiederhergestellte Kopie verfügbar ist ein.

Wählen Sie bei Abrufstufe die zu verwendende Stufe aus.

9. Wählen Sie Weiter aus.
10. Füllen Sie auf der Seite Zusätzliche Optionen konfigurieren die folgenden Abschnitte aus:

- Geben Sie im Abschnitt Zusätzliche Optionen eine Beschreibung des Auftrags und eine Prioritätsnummer für den Auftrag an. Höhere Nummern bedeuten eine höhere Priorität. Weitere Informationen finden Sie unter [the section called “Zuweisen der Auftragspriorität”](#).
- Wählen Sie im Abschnitt Abschlussbericht aus, ob Batch Operations einen Abschlussbericht erstellen soll. Weitere Informationen zu den Fertigstellungsberichten finden Sie unter [the section called “Abschlussberichte”](#).
- Im Abschnitt Berechtigungen müssen Sie Amazon S3 die Erlaubnis erteilen, Batch Operations in Ihrem Namen durchzuführen. Die erforderlichen Berechtigungen finden Sie unter [the section called “Gewähren von Berechtigungen”](#).
- (Optional) fügen Sie im Abschnitt Auftrags-Tags Tags in Schlüssel-Wert-Paaren hinzu. Weitere Informationen finden Sie unter [the section called “Verwenden von Markierungen”](#).

Wählen Sie Weiter aus, sobald Sie fertig sind.

11. Überprüfen Sie die Einstellungen auf der Seite Review. Wenn Sie Änderungen vornehmen müssen, wählen Sie Previous. Wählen Sie andernfalls Auftrag erstellen.

Weitere Informationen über Batch Operations finden Sie unter [Objekte mit Batch Operations wiederherstellen](#) und [Erstellen eines S3-Batch-Vorgangsauftrags](#).

Überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum

Sie können den Status einer Wiederherstellungsanfrage oder das Ablaufdatum überprüfen, indem Sie die Amazon S3-Konsole, Amazon S3 S3-Ereignisbenachrichtigungen AWS CLI, die oder Amazon S3 verwenden RESTAPI.

Note

Objekte, die aus den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive wiederhergestellt wurden, werden nur für die von Ihnen angegebene Anzahl von Tagen gespeichert. Die folgenden Verfahren geben das Ablaufdatum für diese Kopien zurück. Objekte, die aus den Speicherstufen S3 Intelligent-Tiering Archive Access und Deep Archive Access wiederhergestellt wurden, haben kein Ablaufdatum und werden stattdessen zurück in die Stufe für häufigen Zugriff verschoben.

Verwenden der S3-Konsole

So überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum eines Objekts in der Amazon-S3-Konsole

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie wiederherstellen.
4. Wählen Sie in der Liste Objekte das Objekt aus, das Sie wiederherstellen. Die Seite mit den Objektdetails wird angezeigt.
 - Wenn die Wiederherstellung noch nicht abgeschlossen ist, wird oben auf der Seite der Abschnitt Wiederherstellung in Bearbeitung angezeigt.
 - Wenn die Wiederherstellung abgeschlossen ist, wird oben auf der Seite der Abschnitt Wiederherstellung abgeschlossen angezeigt. Wenn Sie aus S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, werden in diesem Abschnitt auch das Ablaufdatum der Wiederherstellung angezeigt. Amazon S3 entfernt die wiederhergestellte Kopie aus dem archivierten Objekt an diesem Datum.

Verwenden von Amazon S3 S3-Ereignisbenachrichtigungen

Sie können über den Abschluss der Objektwiederherstellung informiert werden, indem Sie die `s3:ObjectRestore:Completed` Aktion mit der Amazon S3 S3-Funktion „Event Notifications“ verwenden. Weitere Informationen zum Aktivieren von Ereignisbenachrichtigungen finden Sie unter [Aktivieren von Benachrichtigungen mithilfe von Amazon SQSSNS, Amazon und AWS Lambda](#). Weitere Informationen zu den verschiedenen ObjectRestore Ereignistypen finden Sie unter [the section called “Unterstützte Ereignistypen für SQSSNS, und Lambda”](#).

Mit dem AWS CLI

Überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum eines Objekts mit dem AWS CLI

Im folgenden Beispiel wird der `head-object` Befehl verwendet, um Metadaten für das Objekt anzuzeigen `dir1/example.obj` im Eimer `amzn-s3-demo-bucket`. Wenn Sie diesen Befehl für ein wiederhergestelltes Objekt ausführen, gibt Amazon S3 zurück, ob die Wiederherstellung noch läuft und (falls zutreffend) das Ablaufdatum.


```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Erwartete Ausgabe (Wiederherstellung läuft):

```
{
  "Restore": "ongoing-request=\"true\"",
  "LastModified": "2020-06-16T21:55:22+00:00",
  "ContentLength": 405,
  "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
  "VersionId": "wbYaE2vt0V0iIBXr0qGAJt3fP1cHB8Wi",
  "ContentType": "binary/octet-stream",
  "ServerSideEncryption": "AES256",
  "Metadata": {},
  "StorageClass": "GLACIER"
}
```

Erwartete Ausgabe (Wiederherstellung abgeschlossen):

```
{
  "Restore": "ongoing-request=\"false\", expiry-date=\"Wed, 12 Aug 2020 00:00:00 GMT\"",
  "LastModified": "2020-06-16T21:55:22+00:00",
  "ContentLength": 405,
  "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
  "VersionId": "wbYaE2vt0V0iIBXr0qGAJt3fP1cHB8Wi",
  "ContentType": "binary/octet-stream",
  "ServerSideEncryption": "AES256",
  "Metadata": {},
  "StorageClass": "GLACIER"
}
```

Weitere Informationen zu `head-object` finden Sie [head-object](#) in der AWS CLI Befehlsreferenz.

Mit dem REST API

Amazon S3 bietet Ihnen einen API Vorgang zum Abrufen von Objektmetadaten. Informationen zum Überprüfen des Wiederherstellungsstatus und des Ablaufdatums eines archivierten Objekts mithilfe von finden Sie [HeadObject](#) in der Amazon Simple Storage Service API Reference. REST API

Upgraden der Geschwindigkeit einer Wiederherstellung in Bearbeitung

Sie können die Geschwindigkeit einer Wiederherstellung in Bearbeitung upgraden.

So upgraden Sie eine Wiederherstellung in Bearbeitung auf eine schnellere Stufe:

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie wiederherstellen möchten.
4. Wählen Sie in der Liste Objekte das Objekt aus, das Sie wiederherstellen. Die Seite mit den Objektdetails wird angezeigt. Wählen Sie auf der Detailseite des Objekts Upgrade-Abrufstufe. Informationen zum Überprüfen des Wiederherstellungsstatus eines Objekts finden Sie unter [Überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum](#).
5. Wählen Sie die Stufe aus, auf die Sie upgraden möchten, und dann Wiederherstellen starten aus.

Verwenden der S3-Objektsperre

Mit S3 Object Lock können Sie für einen bestimmten Zeitraum oder auf unbestimmte Zeit verhindern, dass Amazon-S3-Objekte gelöscht oder überschrieben werden. Object Lock verwendet ein write-once-read-many(WORM) -Modell zum Speichern von Objekten. Sie können Object Lock verwenden, um gesetzliche Anforderungen zu erfüllen, die WORM Speicherplatz erfordern, oder um eine weitere Schutzebene gegen Objektänderungen oder das Löschen von Objekten hinzuzufügen.

Note

S3 Object Lock wurde bewertet von Cohasset Associates für den Einsatz in Umgebungen, die den Vorschriften SEC 17a-4, CFTC, und FINRA unterliegen. Weitere Informationen darüber, wie Object Lock mit diesen Vorschriften zusammenhängt, finden Sie im [Cohasset Associates Bewertung der Einhaltung der Vorschriften](#).

Die Objektsperre stellt zwei Optionen für die Verwaltung der Aufbewahrung von Objekten bereit: Aufbewahrungszeiträume und rechtliche Aufbewahrungsfristen. Für eine Objektversion kann es sowohl einen Aufbewahrungszeitraum als auch eine rechtliche Aufbewahrungsfrist geben oder beides geben.

- Aufbewahrungszeitraum – Ein Aufbewahrungszeitraum gibt einen festen Zeitraum an, für den ein Objekt gesperrt bleibt. Sie können einen eindeutigen Aufbewahrungszeitraum für einzelne Objekte festlegen. Darüber hinaus können Sie eine Standardaufbewahrungsdauer für einen S3-

Bucket festlegen. Sie können auch die minimalen und maximalen Aufbewahrungsfristen mit dem `s3:object-lock-remaining-retention-days` Bedingungsschlüssel in der Bucket-Richtlinie einschränken. Auf diese Weise können Sie eine Reihe von Aufbewahrungsfristen festlegen und Aufbewahrungsfristen einschränken, die kürzer oder länger als dieser Zeitraum sein können.

- Gesetzliche Aufbewahrungsfrist – Eine gesetzliche Aufbewahrungsfrist bietet denselben Schutz wie ein Aufbewahrungszeitraum, hat jedoch kein Ablaufdatum. Stattdessen bleibt eine rechtliche Aufbewahrungsfrist gültig, bis Sie diese ausdrücklich entfernen. Rechtliche Aufbewahrungsfristen sind unabhängig von Aufbewahrungsfristen und gelten für einzelne Objektversionen.

Object Lock funktioniert nur in Buckets, für die S3 Versioning aktiviert ist. Wenn Sie eine Objektversion sperren, speichert Amazon S3 die Sperrinformationen in den Metadaten für diese Objektversion. Die Platzierung eines Aufbewahrungszeitraums oder einer rechtlichen Aufbewahrungsfrist für ein Objekt schützt ausschließlich die in der Anforderung angegebene Version. Aufbewahrungsfristen und gesetzliche Aufbewahrungsfristen verhindern nicht, dass neue Versionen des Objekts erstellt werden, und löschen auch keine Markierungen, die über dem Objekt angebracht werden sollen. Weitere Informationen über S3 Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Wenn Sie ein Objekt in einem Bucket platzieren, der bereits ein vorhandenes, geschütztes Objekt mit demselben Objektschlüsselnamen enthält, erstellt Amazon S3 eine neue Version dieses Objekts. Die vorhandene, geschützte Version des Objekts bleibt entsprechend seiner Aufbewahrungskonfiguration gesperrt.

So funktioniert die S3-Objektsperre

Themen

- [Aufbewahrungszeiträume](#)
- [Aufbewahrungsmodi](#)
- [Rechtliche Aufbewahrungsfristen](#)
- [Bewährte Methoden für die Verwendung von S3 Object Lock](#)
- [Erforderliche Berechtigungen](#)

Aufbewahrungszeiträume

Ein Aufbewahrungszeitraum schützt eine Objektversion für einen festen Zeitraum. Wenn Sie für eine Objektversion einen Aufbewahrungszeitraum festlegen, speichert Amazon S3 einen Zeitstempel in

den Metadaten der Objektversion, um anzugeben, wann der Aufbewahrungszeitraum abläuft. Nach Ablauf des Aufbewahrungszeitraums kann die Objektversion überschrieben oder gelöscht werden.

Sie können eine Aufbewahrungsfrist explizit für eine einzelne Objektversion oder für die Eigenschaften eines Buckets festlegen, so dass sie automatisch für alle Objekte in dem Bucket gilt. Wenn Sie einen Aufbewahrungszeitraum explizit auf eine Objektversion anwenden, geben Sie ein Retain Until Date (Bis-Aufbewahrungsdatum) für die Objektversion an. Amazon S3 speichert dieses Datum in den Metadaten der Objektversion.

Sie können auch eine Aufbewahrungsfrist in den Eigenschaften eines Buckets festlegen. Wenn Sie eine Aufbewahrungsfrist für einen Bucket festlegen, geben Sie eine Dauer in Tagen oder Jahren an, für die jede im Bucket platzierte Objektversion geschützt werden soll. Wenn Sie ein Objekt in den Bucket legen, berechnet Amazon S3 ein Bis-Aufbewahrungsdatum für die Objektversion, indem es die angegebene Dauer dem Zeitstempel der Erstellung der Objektversion hinzufügt. Die Objektversion wird anschließend so geschützt, als ob Sie für die Objektversion ausdrücklich eine individuelle Sperre mit diesem Aufbewahrungszeitraum angegeben hätten.

Note

Wenn Sie PUT für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket verwenden, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.

Wie alle anderen Einstellungen für die Objektsperre gelten Aufbewahrungszeiträume für einzelne Objektversionen. Für verschiedene Versionen desselben Objekts können verschiedene Aufbewahrungsmodi und -zeiträume gelten.

z. B.: Angenommen, Sie haben ein Objekt, von dessen 30-tägigem Aufbewahrungszeitraum 15 Tage abgelaufen sind, und Sie führen die Aktion PUT für ein Objekt in Amazon S3 mit demselben Namen und einem 60-tägigen Aufbewahrungszeitraum durch. In diesem Fall ist die PUT-Anforderung erfolgreich und Amazon S3 erstellt eine neue Version des Objekts mit einem 60-tägigen Aufbewahrungszeitraum. Die ältere Version behält den ursprünglichen Aufbewahrungszeitraum und kann in 15 Tagen gelöscht werden.

Sie können einen Aufbewahrungszeitraum verlängern, nachdem Sie eine Aufbewahrungseinstellung auf eine Objektversion angewendet haben. Senden Sie dazu eine neue Object-Lock-Anforderung für die Objektversion mit einem Aufbewahren bis-Datum, das später liegt als das derzeit für

die Objektversion konfigurierte Datum. Amazon S3 ersetzt die bestehende Aufbewahrungsfrist durch den neuen, längeren Zeitraum. Alle Benutzer, die die Berechtigung zur Festlegung von Aufbewahrungszeiträumen für Objekte besitzen, können Aufbewahrungszeiträume für Objektversionen verlängern. Um Aufbewahrungszeitraum festzulegen, müssen Sie die `s3:PutObjectRetention`-Berechtigung besitzen.

Wenn Sie einen Aufbewahrungszeitraum für ein Objekt oder einen S3-Bucket festlegen, müssen Sie einen von zwei Aufbewahrungsmodi auswählen: Compliance oder Governance.

Aufbewahrungsmodi

S3 Object Lock bietet zwei Aufbewahrungsmodi mit unterschiedlichen Schutzebenen für Ihre Objekte:

- Compliance-Modus
- Governance-Modus

Im Compliance-Modus kann eine geschützte Objektversion von keinem Benutzer überschrieben oder gelöscht werden, auch nicht vom Root-Benutzer in Ihrem AWS-Konto. Wenn ein Objekt im Konformitätsmodus gesperrt ist, kann sein Aufbewahrungsmodus nicht geändert werden, und seine Aufbewahrungsdauer kann nicht verkürzt werden. Der Compliance-Modus stellt sicher, dass eine Objektversion während des Aufbewahrungszeitraums weder überschrieben noch gelöscht werden.

Note

Die einzige Möglichkeit, ein Objekt im Konformitätsmodus vor Ablauf seines Aufbewahrungsdatums zu löschen, besteht darin, das zugehörige Objekt zu löschen AWS-Konto.

Im Governance-Modus können Benutzer eine Objektversion nicht überschreiben oder löschen oder ihre Sperreinstellungen ändern, wenn sie keine speziellen Berechtigungen besitzen. Mit dem Governance-Modus schützen Sie Objekte dagegen, von den meisten Benutzern gelöscht zu werden, Sie können jedoch weiterhin einigen Benutzern die Berechtigung geben, die Aufbewahrungseinstellungen zu ändern oder die Objekte bei Bedarf zu löschen. Sie können den Governance-Modus auch verwenden, um die Einstellungen für Aufbewahrungszeiträume zu testen, bevor Sie einen Aufbewahrungszeitraum im Compliance-Modus erstellen.

Um Aufbewahrungseinstellungen im Governance-Modus zu überschreiben oder zu entfernen, müssen Sie die Berechtigung `s3:BypassGovernanceRetention` besitzen und in jede Anfrage, die ein Überschreiben des Governance-Modus erfordert, ausdrücklich `x-amz-bypass-governance-retention:true` als Anfrage-Header einfügen.

Note

Die Amazon-S3-Konsole enthält standardmäßig den `x-amz-bypass-governance-retention:true`-Header. Wenn Sie versuchen, Objekte zu löschen, die durch den Governance-Modus geschützt sind und über die `s3:BypassGovernanceRetention`-Berechtigung verfügen, verläuft der Vorgang erfolgreich.

Rechtliche Aufbewahrungsfristen

Mit Object Lock können Sie auch eine gesetzliche Aufbewahrungsfrist für eine Objektversion festlegen. Wie Aufbewahrungszeiträume verhindern auch rechtliche Aufbewahrungsfristen das Überschreiben oder Löschen von Objektversionen. Mit gesetzlichen Aufbewahrungsfristen sind jedoch keine Aufbewahrungszeiträume verknüpft. Sie sind gültig, bis sie entfernt werden. Alle Benutzer mit der Berechtigung `s3:PutObjectLegalHold` können rechtliche Aufbewahrungsfristen festlegen und entfernen.

Rechtliche Aufbewahrungsfristen sind von Aufbewahrungszeiträumen unabhängig. Die Festlegung einer rechtlichen Aufbewahrungsfrist für eine Objektversion hat keine Auswirkungen auf den Aufbewahrungsmodus oder -zeitraum für die betreffende Objektversion.

Angenommen, Sie setzen eine gesetzliche Aufbewahrungsfrist für eine Objektversion fest, während diese Objektversion gleichzeitig durch einen Aufbewahrungszeitraum geschützt ist. Wenn die Aufbewahrungsfrist abläuft, verliert das Objekt seinen WORM Schutz nicht. Stattdessen wird das Objekt durch die gesetzliche Aufbewahrungsfrist weiter geschützt, bis ein autorisierter Benutzer diese ausdrücklich entfernt. Wenn Sie eine rechtliche Aufbewahrungsfrist entfernen, während für eine Objektversion ein Aufbewahrungszeitraum gilt, bleibt die Objektversion geschützt, bis der Aufbewahrungszeitraum abläuft.

Bewährte Methoden für die Verwendung von S3 Object Lock

Ziehen Sie den Governance-Modus in Betracht, wenn Sie verhindern möchten, dass Objekte während eines vordefinierten Aufbewahrungszeitraums von den meisten Benutzern gelöscht werden,

aber gleichzeitig möchten, dass einige Benutzer mit speziellen Berechtigungen die Flexibilität haben, die Aufbewahrungseinstellungen zu ändern oder die Objekte zu löschen.

Erwägen Sie, den Compliance-Modus zu verwenden, wenn Sie keinen Benutzer haben möchten, auch nicht den Root-Benutzer AWS-Konto, um die Objekte während eines vordefinierten Aufbewahrungszeitraums löschen zu können. Sie können diesen Modus verwenden, falls Sie konforme Daten speichern müssen.

Sie können Legal Hold verwenden, wenn Sie sich nicht sicher sind, wie lange Ihre Objekte unveränderlich bleiben sollen. Das könnte daran liegen, dass Sie eine bevorstehende externe Prüfung Ihrer Daten haben und Objekte bis zum Abschluss der Prüfung unveränderlich lassen möchten. Alternativ haben Sie möglicherweise ein laufendes Projekt, das einen Datensatz verwendet, den Sie bis zum Abschluss des Projekts unveränderlich behalten möchten.

Erforderliche Berechtigungen

Objektsperrenvorgänge erfordern bestimmte Berechtigungen. Abhängig von dem genauen Vorgang, den Sie versuchen, benötigen Sie möglicherweise eine der folgenden Berechtigungen:

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`
- `s3:GetObjectRetention`
- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Eine vollständige Liste der Amazon S3-Berechtigungen mit Beschreibungen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Informationen zur Verwendung von Bedingungen mit Berechtigungen finden Sie unter [Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln](#).

Überlegungen zu Object Lock

Amazon S3 Object Lock kann verhindern, dass Objekte für einen bestimmten Zeitraum oder auf unbestimmte Zeit gelöscht oder überschrieben werden.

Sie können die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), AWS SDKs, oder Amazon S3, REST API um Object Lock-Informationen anzuzeigen oder festzulegen. Weitere Informationen zu den Funktionen von S3 Object Lock finden Sie unter [Verwenden der S3-Objektsperre](#).

Important

- Wenn Sie Object Lock für einen Bucket aktiviert haben, können Sie Object Lock nicht deaktivieren oder die Versionsverwaltung für diesen Bucket aussetzen.
- S3-Buckets mit Object Lock können nicht als Ziel-Buckets für Server-Zugriffsprotokolle verwendet werden. Weitere Informationen finden Sie unter [the section called "Protokollierungs-Serverzugriff"](#).

Themen

- [Berechtigungen zum Anzeigen von Sperrinformationen](#)
- [Umgehen des Governance-Modus](#)
- [Verwenden von Object Lock mit der S3-Replikation](#)
- [Verwenden von Object Lock mit Amazon S3 Inventory](#)
- [Verwaltung von S3-Lebenszyklusrichtlinien mit Object Lock](#)
- [Verwaltung von Löschmarkierungen mit Object Lock](#)
- [Verwenden von S3 Storage Lens mit Object Lock](#)
- [Objekte in einen Bucket mit aktivierter Objektsperre hochladen](#)
- [Konfigurieren von Ereignissen und Benachrichtigungen](#)
- [Festlegen von Beschränkungen für Aufbewahrungsfristen mit einer Bucket-Richtlinie](#)

Berechtigungen zum Anzeigen von Sperrinformationen

Sie können den Object Lock-Status einer Amazon S3 S3-Objektversion programmgesteuert anzeigen, indem Sie den [HeadObject](#) oder [GetObject](#) Operationen. Beide Vorgänge geben den

Aufbewahrungsmodus, die Aufbewahrungsfrist und den Status in Bezug auf eine rechtliche Aufbewahrungsfrist für die angegebene Objektversion an. Darüber hinaus können Sie mithilfe von S3 Inventory den Status der Objektsperre für mehrere Objekte in Ihrem S3-Bucket anzeigen.

Um Aufbewahrungsmodus und Aufbewahrungszeitraum für eine Objektversion anzuzeigen, müssen Sie die Berechtigung `s3:GetObjectRetention` besitzen. Um den Status eines Objekts in Bezug auf rechtliche Aufbewahrungsfristen anzuzeigen, müssen Sie die Berechtigung `s3:GetObjectLegalHold` besitzen. Um die Standardaufbewahrungskonfiguration eines Buckets anzuzeigen, benötigen Sie die `s3:GetBucketObjectLockConfiguration`-Berechtigung. Wenn Sie eine Object-Lock-Konfiguration für einen Bucket anfordern, für den S3 Object Lock nicht aktiviert ist, gibt Amazon S3 einen Fehler zurück.

Umgehen des Governance-Modus

Sie können für im Governance-Modus gesperrte Objektversionen, Vorgänge ausführen, als ob sie nicht geschützt wären, wenn Sie die `s3:BypassGovernanceRetention`-Berechtigung besitzen. Zu diesen Vorgängen gehören das Löschen einer Objektversion, das Verkürzen des Aufbewahrungszeitraums oder das Entfernen des Object-Lock-Aufbewahrungszeitraums durch die Platzierung einer neuen `PutObjectRetention`-Anforderung mit leeren Parametern.

Um den Governance-Modus umgehen zu können, müssen Sie in Ihrer Anforderung ausdrücklich angeben, dass Sie den Governance-Modus umgehen möchten. Fügen Sie dazu den `x-amz-bypass-governance-retention:true` Header Ihrer `PutObjectRetention` API Operationsanforderung hinzu oder verwenden Sie den entsprechenden Parameter bei Anfragen, die über die AWS CLI or AWS SDKs. Die S3-Konsole übernimmt diesen Header automatisch für über die S3-Konsole gestellte Anforderungen, wenn Sie über die `s3:BypassGovernanceRetention`-Berechtigung verfügen, den Governance-Modus zu umgehen.

Note

Die Umgehung des Governance-Modus hat keine Auswirkungen auf den Status einer Objektversion in Bezug auf rechtliche Aufbewahrungsfristen. Wenn für eine Objektversion eine rechtliche Aufbewahrungsfrist aktiviert ist, bleibt diese in Kraft und verhindert, dass die Objektversion durch Anforderungen überschrieben oder gelöscht wird.

Verwenden von Object Lock mit der S3-Replikation

Sie können Object Lock mit S3-Replikation verwenden, um automatisches asynchrones Kopieren von gesperrten Objekten und deren Aufbewahrungs-Metadaten über S3-Buckets hinweg zu aktivieren. Das bedeutet, dass Amazon S3 für replizierte Objekte die Objektsperrekonfiguration des Quell-Buckets verwendet. Mit anderen Worten, wenn im Quell-Bucket Object Lock aktiviert ist, muss in den Ziel-Buckets auch Object Lock aktiviert sein. Wenn ein Objekt direkt in den Ziel-Bucket hochgeladen wird (außerhalb der S3-Replikation), verwendet es die für den Ziel-Bucket festgelegte Objektsperre. Wenn Sie Replikation verwenden, werden Objekte aus einem Quell-Bucket in einen oder mehrere Ziel-Bucket(s) repliziert.

Um die Replikation für einen Bucket mit aktivierter Objektsperre einzurichten, können Sie die S3-Konsole verwenden. AWS CLI, Amazon S3 REST API oder AWS SDKs.

Note

Um Object Lock mit der Replikation zu verwenden, müssen Sie zwei zusätzliche Berechtigungen für den S3-Quell-Bucket im AWS Identity and Access Management (IAM) Rolle, mit der Sie die Replikation einrichten. Die zwei neuen Berechtigungen sind `s3:GetObjectRetention` und `s3:GetObjectLegalHold`. Wenn die Rolle über eine `s3:Get*`-Berechtigungsanweisung verfügt, ist die Anforderung dadurch erfüllt. Weitere Informationen finden Sie unter [Berechtigungen für die Live-Replikation einrichten](#). Allgemeine Informationen zur S3-Replikation finden Sie unter [Überblick über das Replizieren von Objekten](#). Beispiele für die Einrichtung der S3-Replikation finden Sie unter [Beispiele für die Konfiguration der Live-Replikation](#).

Verwenden von Object Lock mit Amazon S3 Inventory

Sie können Amazon S3 Inventory so konfigurieren, dass Listen der Objekte in einem S3-Bucket nach einem definierten Zeitplan erstellt werden. Sie können Amazon S3 Inventory so konfigurieren, dass es die folgenden Object-Lock-Metadaten für Ihre Objekte enthält:

- Das Aufbewahrungsdatum
- Der Aufbewahrungsmodus
- Die gesetzliche Aufbewahrungsfrist

Weitere Informationen finden Sie unter [Amazon S3 Inventory](#).

Verwaltung von S3-Lebenszyklusrichtlinien mit Object Lock

Konfigurationen für die Verwaltung des Objektlebenszyklus funktionieren für geschützte Objekte weiterhin normal. Dies schließt die Platzierung von Löschmarkierungen ein. Eine gesperrte Version eines Objekts kann jedoch nicht durch eine S3-Lifecycle-Ablaufrichtlinie gelöscht werden. Object Lock wird unabhängig davon beibehalten, in welcher Speicherklasse sich das Objekt befindet, und während des gesamten S3 Lifecycle wird zwischen Speicherklassen gewechselt.

Weitere Informationen zur Verwaltung von Objektlebenszyklen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Verwaltung von Löschmarkierungen mit Object Lock

Sie können eine geschützte Objektversion zwar nicht löschen, aber eine Löschmarkierung für das betreffende Objekt erstellen. Durch das Setzen einer Löschmarkierung auf einem Objekt wird keine Objektversion gelöscht. Amazon S3 verhält sich dadurch aber zumeist so, als ob das Objekt gelöscht worden wäre. Weitere Informationen finden Sie unter [Arbeiten mit Löschmarkierungen](#).

Note

Löschmarken sind nicht WORM geschützt, unabhängig von der Aufbewahrungsfrist oder der gesetzlichen Aufbewahrungsfrist für das zugrunde liegende Objekt.

Verwenden von S3 Storage Lens mit Object Lock

Wenn Sie Metriken für objektsperrefähige Speicherbytes und die Anzahl der Objekte sehen möchten, können Sie Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können.

Weitere Informationen finden Sie unter [Verwenden von S3 Storage Lens zum Schutz Ihrer Daten](#).

Eine vollständige Liste der Metriken finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Objekte in einen Bucket mit aktivierter Objektsperre hochladen

Der Content-MD5 Header ist für jede Anfrage zum Hochladen eines Objekts mit einer mit Object Lock konfigurierten Aufbewahrungsfrist erforderlich. Mit dem MD5 Digest können Sie die Integrität

Ihres Objekts überprüfen, nachdem Sie es in einen Bucket hochgeladen haben. Nach dem Hochladen des Objekts berechnet Amazon S3 den MD5 Digest des Objekts und vergleicht ihn mit dem von Ihnen angegebenen Wert. Die Anforderung ist nur erfolgreich, wenn die beiden Digests übereinstimmen. Die S3-Konsole fügt diesen Header automatisch hinzu. Sie müssen diesen Header jedoch angeben, wenn Sie den verwenden. [PutObjectAPI](#)

Weitere Informationen finden Sie unter [Inhalte verwenden — MD5 beim Hochladen von Objekten](#).

Konfigurieren von Ereignissen und Benachrichtigungen

Sie können Amazon S3 Event Notifications verwenden, um den Zugriff und die Änderungen an Ihren Object Lock-Konfigurationen und -Daten nachzuverfolgen, indem Sie AWS CloudTrail. Informationen zu finden CloudTrail Sie unter [Was ist AWS CloudTrail?](#) in der AWS CloudTrail Benutzerleitfaden.

Sie können Amazon auch verwenden CloudWatch , um Benachrichtigungen auf der Grundlage dieser Daten zu generieren. Informationen zu CloudWatch finden Sie unter [Was ist Amazon CloudWatch?](#) im CloudWatch Amazon-Benutzerhandbuch.

Festlegen von Beschränkungen für Aufbewahrungsfristen mit einer Bucket-Richtlinie

Sie können mittels einer Bucket-Richtlinie mindestens erforderliche und maximal zulässige Aufbewahrungszeiträume für einen Bucket festlegen. Die maximale Aufbewahrungsfrist beträgt 100 Jahre.

Das folgende Beispiel zeigt eine Bucket-Richtlinie, die den Bedingungsschlüssel `s3:object-lock-remaining-retention-days` verwendet, um einen maximalen Aufbewahrungszeitraum von 10 Tagen festzulegen.

```
{
  "Version": "2012-10-17",
  "Id": "SetRetentionLimits",
  "Statement": [
    {
      "Sid": "SetRetentionPeriod",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:PutObjectRetention"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
      "Condition": {
        "NumericGreaterThan": {
```

```
    "s3:object-lock-remaining-retention-days": "10"  
  }  
}
```

Note

Wenn es sich bei Ihrem Bucket um den Ziel-Bucket einer Replikationsrichtlinie handelt, können Sie minimal und maximal zulässige Aufbewahrungszeiträume für Objektreplicate einrichten möchten, die mittels Replikation erstellt werden. Hierzu müssen Sie die `s3:ReplicateObject`-Aktion in Ihrer Bucket-Richtlinie zulassen. Weitere Informationen zur Verwendung von Replikationsberechtigungen finden Sie unter [the section called “Einrichten von Berechtigungen”](#).

Weitere Informationen zu Bucket-Richtlinien finden Sie in den folgenden Themen:

- [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

- [Objektoperationen](#)
- [Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln](#)

Konfigurieren von S3 Object Lock

Mit Amazon S3 Object Lock können Sie Objekte in Amazon S3 speichern, indem Sie ein write-once-read-many(WORM) -Modell verwenden. Mit der S3-Objektsperre können Sie für einen festen Zeitraum oder auf unbegrenzte Zeit verhindern, dass ein Objekt gelöscht oder überschrieben wird. Allgemeine Informationen zu den Funktionen von Object Lock finden Sie unter [Verwenden der S3-Objektsperre](#).

Bevor Sie Objekte sperren können, müssen Sie die S3-Versionsverwaltung und Object Lock für einen Bucket aktivieren. Anschließend können Sie einen Aufbewahrungszeitraum, eine gesetzliche Aufbewahrungsfrist oder beides festlegen.

Um mit Object Lock zu arbeiten, müssen Sie über bestimmte Berechtigungen verfügen. Eine Liste der Berechtigungen für verschiedene Object-Lock-Vorgänge finden Sie unter [the section called "Erforderliche Berechtigungen"](#).

Important

- Wenn Sie Object Lock für einen Bucket aktiviert haben, können Sie Object Lock nicht deaktivieren oder die Versionsverwaltung für diesen Bucket aussetzen.
- S3-Buckets mit Object Lock können nicht als Ziel-Buckets für Server-Zugriffsprotokolle verwendet werden. Weitere Informationen finden Sie unter [the section called "Protokollierungs-Serverzugriff"](#).

Themen

- [Aktivieren von Object Lock beim Erstellen eines neuen S3-Buckets](#)
- [Aktivieren von Object Lock für einen vorhandenen S3-Bucket](#)
- [Einrichten oder Ändern einer gesetzlichen Aufbewahrungsfrist für ein S3-Objekt](#)
- [Einrichten oder Ändern eines Aufbewahrungszeitraums für ein S3-Objekt](#)
- [Einrichten oder Ändern eines Standard-Aufbewahrungszeitraums für einen S3-Bucket](#)

Aktivieren von Object Lock beim Erstellen eines neuen S3-Buckets

Sie können Object Lock aktivieren, wenn Sie einen neuen S3-Bucket mithilfe der Amazon S3 S3-Konsole erstellen. AWS Command Line Interface (AWS CLI), AWS SDKs, oder Amazon S3 RESTAPI.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

Anschließend wird die Seite Bucket erstellen geöffnet.

4. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

Note

Der Name eines einmal erstellten Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

5. Wählen Sie für Region AWS-Region wo sich der Bucket befinden soll.
6. Wählen Sie unter Objekteigentümer aus, ob Sie Zugriffskontrolllisten (ACLs) deaktivieren oder aktivieren und den Besitz von Objekten kontrollieren möchten, die in Ihren Bucket hochgeladen wurden.
7. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.
8. Wählen Sie unter Bucket-Versionsverwaltung die Option Aktiviert.

Object Lock funktioniert nur mit versionsgesteuerten Buckets.

9. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher und zur Zuweisung von Kosten verwendet werden.
10. Suchen Sie unter Erweiterte Einstellungen nach Object Lock und wählen Sie Aktivieren aus.

Sie müssen sich darüber im Klaren sein, dass durch die Aktivierung von Object Lock Objekte in diesem Bucket dauerhaft gesperrt werden können.

11. Wählen Sie Bucket erstellen aus.

Verwendung der AWS CLI

Im folgenden `create-bucket`-Beispiel wird ein neuer S3-Bucket mit dem Namen *amzn-s3-demo-bucket1* und aktiviertem Object Lock erstellt:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket1 --object-lock-enabled-for-bucket
```

Weitere Informationen und Beispiele finden Sie unter [create-bucket](#) in der AWS CLI Befehlsreferenz.

Note

Du kannst rennen AWS CLI Befehle von der Konsole aus mit AWS CloudShell. AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von AWS Management Console. Weitere Informationen finden Sie unter [Was ist CloudShell?](#) in der AWS CloudShell Benutzerleitfaden.

Mit dem REST API

Sie können den verwenden RESTAPI, um einen neuen S3-Bucket mit aktivierter Objektsperre zu erstellen. Weitere Informationen finden Sie unter [CreateBucket](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Hier finden Sie Beispiele dafür, wie Sie Object Lock aktivieren, wenn Sie einen neuen S3-Bucket mit dem AWS SDKs, siehe [Verwenden Sie CreateBucket mit einem AWS SDKoder CLI](#).

Beispiele dafür, wie Sie die aktuelle Object Lock-Konfiguration mit dem AWS SDKs, siehe [Verwenden Sie GetObjectLockConfiguration mit einem AWS SDKoder CLI](#).

Ein interaktives Szenario zur Demonstration verschiedener Object Lock-Funktionen finden Sie unter AWS SDKs, siehe [Arbeiten Sie mit den Objektsperrefunktionen von Amazon S3 mithilfe eines AWS SDK](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Aktivieren von Object Lock für einen vorhandenen S3-Bucket

Sie können Object Lock für einen vorhandenen S3-Bucket aktivieren, indem Sie die Amazon S3 S3-Konsole verwenden, AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI.

Verwenden der S3-Konsole

Note

Object Lock funktioniert nur mit versionsgesteuerten Buckets.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Object Lock aktivieren möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie unter Eigenschaften nach unten zum Bereich Object Lock und wählen Sie Bearbeiten aus.
6. Wählen Sie unter Object Lock die Option Aktivieren aus.

Sie müssen sich darüber im Klaren sein, dass durch die Aktivierung von Object Lock Objekte in diesem Bucket dauerhaft gesperrt werden können.

7. Wählen Sie Änderungen speichern.

Verwendung der AWS CLI

Mit dem folgenden `put-object-lock-configuration`-Beispielbefehl wird eine Aufbewahrungsfrist von 50 Tagen für Object Lock für einen Bucket mit dem Namen *amzn-s3-demo-bucket1* festgelegt:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Weitere Informationen und Beispiele finden Sie unter [put-object-lock-configuration](#) in der AWS CLI Befehlsreferenz.

Note

Du kannst rennen AWS CLI Befehle von der Konsole aus mit AWS CloudShell. AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von AWS Management Console. Weitere Informationen finden Sie unter [Was ist CloudShell?](#) in der AWS CloudShell Benutzerleitfaden.

Mit dem REST API

Sie können Amazon S3 verwenden RESTAPI, um Object Lock für einen vorhandenen S3-Bucket zu aktivieren. Weitere Informationen finden Sie unter [PutObjectLockConfiguration](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Beispiele für die Aktivierung von Object Lock für einen vorhandenen S3-Bucket finden Sie mit AWS SDKs, siehe [Verwenden Sie PutObjectLockConfiguration mit einem AWS SDK oder CLI](#).

Beispiele dafür, wie Sie die aktuelle Object Lock-Konfiguration mit dem AWS SDKs, siehe [Verwenden Sie GetObjectLockConfiguration mit einem AWS SDK oder CLI](#).

Ein interaktives Szenario zur Demonstration verschiedener Object Lock-Funktionen finden Sie unter AWS SDKs, siehe [Arbeiten Sie mit den Objektsperrefunktionen von Amazon S3 mithilfe eines AWS SDK](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Einrichten oder Ändern einer gesetzlichen Aufbewahrungsfrist für ein S3-Objekt

Sie können mithilfe der Amazon S3-Konsole eine gesetzliche Sperre für ein S3-Objekt festlegen oder entfernen. AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI.

Important

- Wenn Sie eine gesetzliche Aufbewahrungsfrist für ein Objekt festlegen möchten, muss Object Lock für den Bucket des Objekts bereits aktiviert sein.
- Wenn Sie PUT für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket verwenden, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.

Weitere Informationen finden Sie unter [the section called "Rechtliche Aufbewahrungsfristen"](#).

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält, für das Sie eine gesetzliche Aufbewahrungsfrist einrichten oder ändern möchten.
4. Wählen Sie in der Liste Objekte das Objekt aus, für das Sie eine gesetzliche Aufbewahrungsfrist festlegen oder ändern möchten.
5. Suchen Sie auf der Seite mit den Objekteigenschaften den Bereich Object Lock – Gesetzliche Aufbewahrungsfristen und wählen Sie Bearbeiten.
6. Wählen Sie Aktivieren, um eine gesetzliche Aufbewahrungsfrist festzulegen, oder Deaktivieren, um eine gesetzliche Aufbewahrungsfrist zu entfernen.
7. Wählen Sie Änderungen speichern.

Verwendung der AWS CLI

Im folgenden `put-object-legal-hold` Beispiel wird ein legaler Hold-Wert für das Objekt festgelegt *my-image.fs* in dem Bucket mit dem Namen: *amzn-s3-demo-bucket1*

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --legal-hold="Status=ON"
```

Im folgenden `put-object-legal-hold` Beispiel wird eine gesetzliche Sperre für das Objekt aufgehoben *my-image.fs* in dem Bucket mit dem Namen: *amzn-s3-demo-bucket1*

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --legal-hold="Status=OFF"
```

Weitere Informationen und Beispiele finden Sie unter [put-object-legal-hold](#) in der AWS CLI Befehlsreferenz.

Note

Du kannst rennen AWS CLI Befehle von der Konsole aus mit AWS CloudShell. AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von AWS

Management Console. Weitere Informationen finden Sie unter [Was ist CloudShell?](#) in der AWS CloudShell Benutzerleitfaden.

Mit dem REST API

Sie können den verwenden RESTAPI, um eine gesetzliche Sperre für ein Objekt festzulegen oder zu ändern. Weitere Informationen finden Sie unter [PutObjectLegalHold](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Hier finden Sie Beispiele dafür, wie Sie einen gesetzlichen Aufbewahrungszeitraum für ein Objekt festlegen AWS SDKs, siehe [Verwenden Sie PutObjectLegalHold mit einem AWS SDKoder CLI](#).

Beispiele dafür, wie Sie den aktuellen Status der gesetzlichen Aufbewahrungsfrist erhalten können, finden Sie unter AWS SDKs, siehe [Verwenden Sie GetObjectLegalHold mit einem AWS SDKoder CLI](#).

Ein interaktives Szenario zur Demonstration verschiedener Object Lock-Funktionen finden Sie unter AWS SDKs, siehe [Arbeiten Sie mit den Objektsperreffunktionen von Amazon S3 mithilfe eines AWS SDK](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Einrichten oder Ändern eines Aufbewahrungszeitraums für ein S3-Objekt

Sie können einen Aufbewahrungszeitraum für ein S3-Objekt mithilfe der Amazon S3 S3-Konsole festlegen oder ändern. AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI.

Important

- Wenn Sie einen Aufbewahrungszeitraum für ein Objekt festlegen möchten, muss Object Lock für den Bucket des Objekts bereits aktiviert sein.
- Wenn Sie PUT für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket verwenden, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.

- Die einzige Möglichkeit, ein Objekt im Compliance-Modus vor Ablauf seines Aufbewahrungsdatums zu löschen, besteht darin, das zugehörige Objekt zu löschen AWS-Konto.

Weitere Informationen finden Sie unter [Aufbewahrungszeiträume](#).

Verwenden der S3-Konsole


1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält, für das Sie einen Aufbewahrungszeitraum einrichten oder ändern möchten.
4. Wählen Sie in der Liste Objekte das Objekt aus, für das Sie einen Aufbewahrungszeitraum festlegen oder ändern möchten.
5. Suchen Sie auf der Seite mit den Objekteigenschaften den Bereich Object Lock – Aufbewahrungszeitraum und wählen Sie Bearbeiten.
6. Wählen Sie unter Aufbewahrung die Option Aktivieren aus, um einen Aufbewahrungszeitraum festzulegen, oder Deaktivieren, um einen Aufbewahrungszeitraum zu entfernen.
7. Wenn Sie Aktivieren unter Aufbewahrungsmodus ausgewählt haben, wählen Sie entweder Governance-Modus oder Compliance-Modus. Weitere Informationen finden Sie unter [Aufbewahrungsmodi](#).
8. Wählen Sie unter Aufbewahren bis das Datum aus, an dem der Aufbewahrungszeitraum enden soll. Während dieses Zeitraums ist Ihr Objekt WORM geschützt und kann nicht überschrieben oder gelöscht werden. Weitere Informationen finden Sie unter [Aufbewahrungszeiträume](#).
9. Wählen Sie Save Changes (Änderungen speichern).

Verwendung der AWS CLI

Im folgenden `put-object-retention` Beispiel wird eine Aufbewahrungsfrist für das Objekt festgelegt *my-image.fs* in dem Bucket, der *amzn-s3-demo-bucket1* bis zum 1. Januar 2025 benannt wurde:

```
aws s3api put-object-retention --bucket amzn-s3-demo-bucket1 --key my-image.fs --  
retention='{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Weitere Informationen und Beispiele finden Sie unter [put-object-retention](#) in der AWS CLI Befehlsreferenz.

 Note

Du kannst rennen AWS CLI Befehle von der Konsole aus mit AWS CloudShell. AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von AWS Management Console. Weitere Informationen finden Sie unter [Was ist CloudShell?](#) in der AWS CloudShell Benutzerleitfaden.

Mit dem REST API

Sie können den verwenden RESTAPI, um eine Aufbewahrungsfrist für ein Objekt festzulegen. Weitere Informationen finden Sie unter [PutObjectRetention](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Beispiele dafür, wie Sie eine Aufbewahrungsfrist für ein Objekt festlegen können, finden Sie unter AWS SDKs, siehe [Verwenden Sie PutObjectRetention mit einem AWS SDKoder CLI](#).

Beispiele dafür, wie Sie die Aufbewahrungsdauer für ein Objekt ermitteln können, finden Sie mit AWS SDKs, siehe [Verwenden Sie GetObjectRetention mit einem AWS SDKoder CLI](#).

Ein interaktives Szenario zur Demonstration verschiedener Object Lock-Funktionen finden Sie unter AWS SDKs, siehe [Arbeiten Sie mit den Objektsperrefunktionen von Amazon S3 mithilfe eines AWS SDK](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Einrichten oder Ändern eines Standard-Aufbewahrungszeitraums für einen S3-Bucket

Sie können mithilfe der Amazon S3 S3-Konsole eine Standardaufbewahrungsdauer für einen S3-Bucket festlegen oder ändern. AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI. Sie geben eine Dauer in Tagen oder Jahren an, für die jede in dem Bucket befindliche Objektversion geschützt werden soll.

Important

- Wenn Sie einen Standard-Aufbewahrungszeitraum für einen Bucket festlegen möchten, muss Object Lock für den Bucket bereits aktiviert sein.
- Wenn Sie für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket PUT durchführen, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.
- Die einzige Möglichkeit, ein Objekt im Compliance-Modus vor Ablauf seines Aufbewahrungsdatums zu löschen, besteht darin, das zugehörige Objekt zu löschen AWS-Konto.

Weitere Informationen finden Sie unter [Aufbewahrungszeiträume](#).

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie einen Aufbewahrungszeitraum einrichten oder ändern möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie unter Eigenschaften nach unten zum Bereich Object Lock und wählen Sie Bearbeiten aus.
6. Wählen Sie unter Standard-Aufbewahrung die Option Aktivieren aus, um einen Standard-Aufbewahrungszeitraum festzulegen, oder Deaktivieren, um einen Standard-Aufbewahrungszeitraum zu entfernen.
7. Wenn Sie Aktivieren unter Aufbewahrungsmodus ausgewählt haben, wählen Sie entweder Governance-Modus oder Compliance-Modus. Weitere Informationen finden Sie unter [Aufbewahrungsmodi](#).
8. Wählen Sie unter Standard-Aufbewahrungszeitraum die Anzahl der Tage oder Jahre aus, für die der Aufbewahrungszeitraum gelten soll. Objekte, die sich in diesem Bucket befinden, werden für diese Anzahl von Tagen oder Jahren gesperrt. Weitere Informationen finden Sie unter [Aufbewahrungszeiträume](#).

9. Wählen Sie Save Changes (Änderungen speichern).

Verwendung der AWS CLI

Mit dem folgenden `put-object-lock-configuration`-Beispielbefehl wird eine Object-Lock-Aufbewahrungsfrist von 50 Tag für einen Bucket mit dem Namen `amzn-s3-demo-bucket1` unter Verwendung des Compliance-Modus festgelegt:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Im folgenden `put-object-lock-configuration`-Beispiel wird die Standard-Aufbewahrungskonfiguration für einen Bucket entfernt:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-lock-configuration='{ "ObjectLockEnabled": "Enabled" }'
```

Weitere Informationen und Beispiele finden Sie unter [put-object-lock-configuration](#) in der AWS CLI Befehlsreferenz.

Note

Du kannst rennen AWS CLI Befehle von der Konsole aus mit AWS CloudShell. AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von AWS Management Console. Weitere Informationen finden Sie unter [Was ist CloudShell?](#) in der AWS CloudShell Benutzerleitfaden.

Mit dem REST API

Sie können den verwenden RESTAPI, um einen Standard-Aufbewahrungszeitraum für einen vorhandenen S3-Bucket festzulegen. Weitere Informationen finden Sie unter [PutObjectLockConfiguration](#) in der Amazon Simple Storage Service API Reference.

Verwendung der AWS SDKs

Beispiele dafür, wie Sie eine standardmäßige Aufbewahrungsfrist für einen vorhandenen S3-Bucket festlegen, finden Sie unter AWS SDKs, siehe [Verwenden Sie PutObjectLockConfiguration mit einem AWS SDKoder CLI](#).

Ein interaktives Szenario zur Demonstration verschiedener Object Lock-Funktionen finden Sie unter [AWS SDKs](#), siehe [Arbeiten Sie mit den Objektsperrefunktionen von Amazon S3 mithilfe eines AWS SDK](#).

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs, siehe [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).

Verwenden von Amazon-S3-Speicherklassen

Jedem Objekt in Amazon S3 ist eine Speicherklasse zugeordnet. Wenn Sie beispielsweise alle Objekte in einem S3-Bucket auflisten, zeigt die Konsole die Speicherklasse für alle Objekte in der Liste an. Amazon S3 bietet eine Reihe von Speicherklassen für die von Ihnen gespeicherten Objekte an. Sie wählen eine Speicherklasse abhängig von Ihrem Anwendungsszenario und den Leistungsanforderungen für den Zugriff aus. Alle diese Speicherklassen bieten höchste Haltbarkeit.

Die folgenden Abschnitte enthalten Details zu den verschiedenen Speicherklassen und zum Festlegen einer Speicherklasse für Ihre Objekte.

Themen


- [Speicherklassen für Objekte mit häufigem Zugriff](#)
- [Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekanntem Zugriffsmustern](#)
- [Speicherklassen für Objekte mit seltenem Zugriff](#)
- [Speicherklassen für Objekte, auf die selten zugegriffen wird](#)
- [Speicherklasse für Amazon S3 in Outposts](#)
- [Vergleich der Amazon-S3-Speicherklassen](#)
- [Einrichten der Speicherklasse eines Objekts](#)

Speicherklassen für Objekte mit häufigem Zugriff

Für leistungssensible Anwendungsfälle (die eine Zugriffszeit von Millisekunden erfordern) und Daten mit häufigem Zugriff stellt Amazon S3 die folgenden Speicherklassen bereit:

- S3 Standard – Die Standardspeicherklasse. Wenn Sie beim Upload eines Objekts keine Speicherklasse angeben, weist Amazon S3 die Speicherklasse S3 Standard zu.

- **S3 Express One Zone** – Amazon S3 Express One Zone ist eine hochleistungsfähige Amazon-S3-Speicherklasse mit einer einzelnen Zone, die entwickelt wurde, um einen konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre Daten und für latenzempfindliche Anwendungen zu gewährleisten. S3 Express One Zone ist die derzeit verfügbare Cloud-Objektspeicherklasse mit der niedrigsten Latenz. Die Datenzugriffsgeschwindigkeit ist bis zu zehnmal schneller und die Anforderungskosten sind 50 Prozent niedriger als bei S3 Standard. Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. Weitere Informationen finden Sie unter [Was ist S3 Express One Zone?](#).
- **Reduzierte Redundanz** — Die Speicherklasse Reduced Redundancy Storage (RRS) wurde für unkritische, reproduzierbare Daten konzipiert, die mit weniger Redundanz gespeichert werden können als die Speicherklasse S3 Standard.

 **Important**

Wir empfehlen, diese Speicherklasse nicht zu verwenden. Die Speicherklasse S3 Standard ist kostengünstiger.

Aus Gründen der Haltbarkeit wird bei RRS Objekten ein durchschnittlicher jährlicher Verlust von 0,01 Prozent der Objekte erwartet. Wenn ein RRS Objekt verloren geht und Anfragen an dieses Objekt gestellt werden, gibt Amazon S3 einen 405-Fehler zurück.

Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekanntem Zugriffsmustern

S3 Intelligent-Tiering ist eine Amazon-S3-Speicherklasse zur Optimierung der Speicherkosten durch automatisches Verschieben von Daten auf die kostengünstigste Zugriffsebene ohne Leistungsauswirkung oder Betriebsaufwand. S3 Intelligent-Tiering ist die einzige Cloud-Speicherklasse, die automatische Kosteneinsparungen durch Verschieben von Daten auf granularer Objektebene zwischen den Zugriffsebenen ermöglicht, wenn sich die Zugriffsmuster ändern. S3 Intelligent-Tiering ist die ideale Speicherklasse, wenn Sie die Speicherkosten für Daten mit unbekanntem oder sich ändernden Zugriffsmustern optimieren möchten. Bei S3 Intelligent-Tiering werden keine Abrufgebühren erhoben.

Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt die Objekte, auf die nicht zugegriffen wurde, automatisch in kostengünstigere Zugriffsebenen. S3 Intelligent-Tiering ermöglicht automatische

Speicherkosteneinsparungen in drei Zugriffsebenen mit niedriger Latenz und hohem Durchsatz. Für Daten, auf die asynchron zugegriffen werden kann, können Sie innerhalb der S3 Intelligent-Tiering-Speicherklasse die automatische Archivfunktionen aktivieren. S3 Intelligent-Tiering ist auf 99,9 % Verfügbarkeit und 99,9999999 % Haltbarkeit ausgelegt.

S3 Intelligent-Tiering speichert Objekte automatisch in drei Zugriffsebenen:

- **Frequent Access (Häufige Zugriffe)** – In S3 Intelligent-Tiering hochgeladene oder übertragene Objekte werden automatisch in der Ebene Frequent Access gespeichert.
- **Infrequent Access (Seltene Zugriffe)** – S3 Intelligent-Tiering verschiebt Objekte, auf die 30 aufeinanderfolgende Tage lang nicht zugegriffen wurde, in die Ebene Infrequent Access.
- **Archive Instant Access (Archiv für sofortigen Zugriff)** – Mit S3 Intelligent-Tiering werden alle vorhandenen Objekte, auf die 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde, automatisch in die Ebene Archive Instant Access verschoben.

Zusätzlich zu diesen drei Stufen bietet S3 Intelligent-Tiering zwei optionale Archivzugriffsebenen:

- **Archive Access (Archivzugriff)** – S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Archivzugriffsebene für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Archive Access automatisch Objekte, auf die mindestens 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde.
- **Deep Archive Access (tiefer Archivzugriff)** – S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Stufe Deep Archive Access für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Deep Archive Access automatisch Objekte, auf die mindestens 180 aufeinanderfolgende Tage lang nicht zugegriffen wurde.

Note

- Aktivieren Sie die Stufe Archive Access nur für 90 Tage, wenn Sie die Stufe Instant Access Archiv umgehen möchten. Die Stufe Archive Access bietet etwas kostengünstigeren Speicherplatz mit minute-to-hour Abrufzeiten. Die Stufe Archive Instant Access bietet Millisekunden-Zugriff und hohe Durchsatzleistung.
- Aktivieren Sie die Ebenen Archivzugriff und Deep Archive Access nur, wenn Ihre Anwendung asynchron auf Ihre Objekte zugreifen kann. Wenn das Objekt, das Sie abrufen,

in den Stufen Archive Access oder Deep Archive Access gespeichert ist, stellen Sie das Objekt zunächst mithilfe von `RestoreObject` wieder her.

Sie können [neu erstellte Daten in S3 Intelligent-Tiering verschieben](#) und dies als Standard-speicherklasse festlegen. Sie können auch wählen, ob Sie eine oder beide Archivzugriffsebenen aktivieren möchten, indem Sie den [PutBucketIntelligentTieringConfiguration](#) API-Vorgang verwenden AWS CLI, oder die Amazon S3 S3-Konsole. Weitere Informationen zur Verwendung von S3 Intelligent-Tiering und zur Aktivierung der Archivzugriffsebenen finden Sie unter [Verwenden von S3 Intelligent-Tiering](#).

Um auf Objekte in den Stufen „Archive Access“ oder „Deep Archive Access“ zugreifen zu können, müssen Sie sie zunächst wiederherstellen. Weitere Informationen finden Sie unter [Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access](#).

Note

Wenn ein Objekt kleiner als 128 KB ist, wird es nicht überwacht und ist nicht für das Auto-Tiering geeignet. Kleinere Objekte werden immer in der Stufe für häufige Zugriffe gespeichert. Weitere Informationen zu S3 Intelligent-Tiering finden Sie unter [S3-Intelligent-Tiering-Zugriffsebenen](#).

Speicherklassen für Objekte mit seltenem Zugriff

Die Speicherklassen S3 Standard-IA und S3 One Zone-IA sind für langlebige Daten, auf die selten zugegriffen wird, konzipiert. (IA steht für Infrequent Access (seltener Zugriff).) S3 Standard-IA- und S3 One Zone-IA-Objekte sind für den Zugriff in Millisekunden verfügbar (ähnlich wie die Speicherklasse S3 Standard). Da Amazon S3 eine Abrufgebühr für diese Objekte erhebt, sind sie am besten für Daten mit seltenem Zugriff geeignet. Informationen zu den Preisen finden Sie unter [Amazon S3 – Preise](#).

Beispielsweise können Sie die Speicherklassen S3 Standard-IA und S3 One Zone-IA für Folgendes auswählen:

- Zum Speichern von Sicherungen.
- Für ältere Daten, auf die selten zugegriffen wird, für die aber dennoch ein Zugriff in Millisekunden erforderlich ist. Beim Upload von Daten können Sie beispielsweise die S3 Standard-Speicherklasse

auswählen und dann Amazon S3 mithilfe der Lebenszyklus-Konfiguration anweisen, die Objekte in die Klasse S3 Standard-IA oder S3 One Zone-IA zu überführen.

Weitere Informationen zur Lebenszyklusverwaltung finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Note

Die Speicherklassen S3 Standard-IA und S3 One Zone-IA sind für Objekte mit mehr als 128 KB geeignet, die Sie für mindestens 30 Tage aufbewahren wollen. Ist ein Objekt kleiner als 128 KB, fallen Amazon-S3-Gebühren für 128 KB an. Wenn Sie ein Objekt vor dem Ablauf der 30-tägigen Mindestspeicherdauer löschen, werden Ihnen 30 Tage berechnet. Für Objekte, die vor Ablauf der 30-Tage-Frist gelöscht, überschrieben oder in eine andere Speicherklasse verschoben werden, wird die übliche nutzungsabhängige Gebühr sowie eine anteilige Gebühr für die restlichen Tage des 30-Tage-Minimums in Rechnung gestellt. Informationen zu den Preisen finden Sie unter [Amazon S3 – Preise](#).

Diese Speicherklassen weisen folgende Unterschiede auf:

- S3 Standard-IA – Amazon S3 speichert die Objektdaten redundant in mehreren geografisch getrennten Availability Zones (ähnlich wie die Speicherklasse S3 Standard). S3 Standard-IA-Objekte sind von einem Availability Zone-Ausfall nicht betroffen. Diese Speicherklasse bietet mehr Verfügbarkeit und Stabilität als die Klasse S3 One Zone-IA.
- S3 One Zone-IA – Amazon S3 speichert die Objektdaten nur in einer Availability Zone, daher ist diese Variante kostengünstiger als S3 Standard-IA. Allerdings sind die Daten bei einem physischen Ausfall der Availability Zone (aufgrund von Katastrophen wie z. B. Erdbeben und Überflutungen) nicht sicher. Die Speicherklasse S3 One Zone-IA bietet dieselbe Zuverlässigkeit wie S3 Standard-IA, ist jedoch weniger verfügbar und weniger ausfallsicher. Einen Vergleich der Zuverlässigkeit und Verfügbarkeit von Speicherklassen finden Sie unter [Vergleich der Amazon-S3-Speicherklassen](#) am Ende dieses Abschnitts. Informationen zu den Preisen finden Sie unter [Amazon S3 – Preise](#).

Wir empfehlen Folgendes:

- S3 Standard-IA – Verwenden Sie diese Klasse für die primäre (oder einzige) Kopie der Daten, die nicht neu erstellt werden können.

- S3 One Zone-IA — Verwenden Sie diese Option, wenn Sie die Daten neu erstellen können, falls die Availability Zone ausfällt, und für Objektreplikate bei der Konfiguration von S3 Cross-Region Replication (). CRR

Speicherklassen für Objekte, auf die selten zugegriffen wird

Die Speicherklassen S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind für die kostengünstige, langfristige Datenspeicherung und Datenarchivierung konzipiert. Diese Speicherklassen bieten dieselbe Zuverlässigkeit und Stabilität wie die Speicherklasse S3 Standard und S3 Standard-IA Speicherklassen. Weitere Informationen zu den S3 Glacier-Speicherklassen finden Sie unter [Langfristige Datenspeicherung mit S3 Glacier-Speicherklassen](#)

Amazon S3 bietet die folgenden S3 Glacier-Speicherklassen:

- S3 Glacier Instant Retrieval — Wird für Langzeitdaten verwendet, auf die selten zugegriffen wird und deren Abruf Millisekunden erfordert. Daten in dieser Speicherklasse sind für den Echtzeitzugriff verfügbar.
- S3 Glacier Flexible Retrieval – Diese Klasse eignet sich für Archive, bei denen Teile der Daten möglicherweise innerhalb von wenigen Minuten abgerufen werden müssen. Daten in dieser Speicherklasse sind archiviert und für den Echtzeitzugriff nicht verfügbar.
- S3 Glacier Deep Archive – Verwenden Sie diese Speicherklasse zur Archivierung von Daten, auf die nur selten zugegriffen werden muss. Daten in dieser Speicherklasse sind archiviert und für den Echtzeitzugriff nicht verfügbar.

Abrufen archivierter Objekte

Sie können die Speicherklasse eines Objekts auf S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive setzen, genau so wie für andere Speicherklassen, wie im Abschnitt [Einrichten der Speicherklasse eines Objekts](#) beschrieben. S3 Glacier Flexible Retrieval- und S3 Glacier Deep Archive Archive-Objekte werden jedoch archiviert und sind nicht für den Echtzeitzugriff verfügbar. Weitere Informationen finden Sie unter [Archivierter Speicher](#).

Note

Wenn Sie S3 Glacier-Speicherklassen verwenden, verbleiben Ihre Objekte in Amazon S3. Sie können nicht direkt über den separaten Amazon-S3-Glacier-Service darauf zugreifen.

Informationen zum Amazon S3 Glacier-Service finden Sie im [Amazon S3 Glacier Developer Guide](#).

Speicherklasse für Amazon S3 in Outposts

Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihrem erstellten AWS Outposts Ressourcen und das Speichern und Abrufen von Objekten vor Ort für Anwendungen, die lokalen Datenzugriff, lokale Datenverarbeitung und Datenresidenz erfordern. Sie können dieselben API Operationen und Funktionen für verwenden AWS Outposts wie Sie es bei Amazon S3 tun, einschließlich Zugriffsrichtlinien, Verschlüsselung und Tagging. Sie können S3 auf Outposts verwenden über AWS Management Console, AWS CLI, AWS SDKs, oder RESTAPI.

S3 in Outposts bietet eine neue Speicherklasse: S3 Outposts (OUTPOSTS). Die Speicherklasse S3 Outposts ist nur für Objekte verfügbar, die in Buckets in Outposts gespeichert sind. Wenn Sie versuchen, diese Speicherklasse mit einem S3-Bucket in einem zu verwenden AWS-Region, ein `InvalidStorageClass` Fehler tritt auf. Sie erhalten dieselbe Fehlermeldung, wenn Sie versuchen, andere S3-Speicherklassen mit Objekten zu verwenden, die in S3-on-Outposts-Buckets gespeichert sind.

Objekte, die in der Speicherklasse S3 Outposts (OUTPOSTS) gespeichert sind, werden immer mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) verschlüsselt. Weitere Informationen finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

Sie können sich auch explizit dafür entscheiden, Objekte, die in der Speicherklasse S3 Outposts gespeichert sind, zu verschlüsseln, indem Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (-C) verwenden. SSE Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln \(-C\) SSE](#).

Note

S3 on Outposts unterstützt keine serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS).

Weitere Informationen zu S3 on Outposts finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Vergleich der Amazon-S3-Speicherklassen

In der folgenden Tabelle werden die Speicherklassen verglichen, einschließlich ihrer Verfügbarkeit, Haltbarkeit, Mindestspeicherdauer und anderer Faktoren.

Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects .
DEEP_ARCHIVE	Archiving rarely accessed data with a default retrieval time of 12 hours	99.999999999%	99.99% (after you restore objects)	>= 3	180 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects .
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

* S3 Glacier Flexible Retrieval erfordert 40 KB zusätzliche Metadaten für jedes archivierte Objekt. Dazu gehören 32 KB Metadaten, die mit dem Tarif von S3 Glacier Flexible Retrieval abgerechnet werden (erforderlich, um Ihre Daten zu identifizieren und abzurufen), und weitere 8 KB Daten, die zum S3-Standardtarif berechnet werden. Der S3-Standardtarif ist erforderlich, um den benutzerdefinierten Namen und die Metadaten für Objekte zu verwalten, die in S3 Glacier Flexible Retrieval archiviert wurden. Weitere Informationen zu Speicherklassen finden Sie unter [Amazon-S3-Speicherklassen](#).

** S3 Glacier Deep Archive benötigt 40 KB an zusätzlichen Metadaten für jedes archivierte Objekt. Dazu gehören 32 KB Metadaten, die mit dem Tarif von S3 Glacier Deep Archive abgerechnet werden (erforderlich, um Ihre Daten zu identifizieren und abzurufen), und weitere 8 KB Daten, die zum S3-Standardtarif berechnet werden. Der S3-Standardtarif ist erforderlich, um den benutzerdefinierten Namen und die Metadaten für Objekte zu verwalten, die in Amazon S3 Glacier Deep Archive archiviert wurden. Weitere Informationen zu Speicherklassen finden Sie unter [Amazon-S3-Speicherklassen](#).

Beachten Sie, dass alle Speicherklassen außer S3 One Zone-IA und S3 Express One Zone gegenüber dem physischen Verlust einer Availability Zone aufgrund von Katastrophen ausfallsicher

sind. Berücksichtigen Sie neben den Leistungsanforderungen Ihres Anwendungsszenarios auch die Kosten. Preisinformationen für Speicherklassen finden Sie unter [Amazon S3 – Preise](#).

Einrichten der Speicherklasse eines Objekts

Um Objektspeicherklassen festzulegen und zu aktualisieren, können Sie die Amazon S3 S3-Konsole verwenden. AWS SDKs, oder die AWS Command Line Interface (AWS CLI). Alle diese Ansätze verwenden Amazon API S3-Operationen, um Anfragen an Amazon S3 zu senden.

Amazon S3 API S3-Operationen unterstützen das Einstellen (oder Aktualisieren) der Speicherklasse von Objekten wie folgt:

- Beim Erstellen eines neuen Objekts können Sie dessen Speicherklasse angeben. Wenn Sie beispielsweise Objekte mithilfe der API Operationen [PUTObject, Object](#) und [POSTInitiate Multipart Upload](#) erstellen, fügen Sie den `x-amz-storage-class` Anforderungsheader hinzu, um eine Speicherklasse anzugeben. Falls Sie diesen Header weglassen, verwendet Amazon S3 die Standardspeicherklasse S3 Standard.
- Sie können auch die Speicherklasse eines Objekts, das bereits in Amazon S3 gespeichert ist, in eine andere Speicherklasse ändern, indem Sie mithilfe der API Operation Objekt [— Kopieren eine Kopie des PUT Objekts](#) erstellen. Sie können jedoch nicht [PUTT Object - Copy](#) verwenden, um in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeicherte Objekte zu kopieren. Sie können auch nicht von S3 One Zone-IA zu S3 Glacier Instant Retrieval wechseln.

Sie kopieren das Objekt unter Verwendung desselben Schlüsselnamens in denselben Bucket und geben die Anforderungs-Header wie folgt an:

- Legen Sie den `x-amz-metadata-directive`-Header auf COPY fest.
- Legen Sie den `x-amz-storage-class`-Header auf die zu verwendende Speicherklasse fest.

In einem Bucket mit aktiviertem Versioning kann die Speicherklasse einer spezifischen Objektversion nicht geändert werden. Beim Kopieren des Objekts vergibt Amazon S3 eine neue Versions-ID.

- Sie können die Speicherklasse eines Objekts mithilfe der Amazon S3 S3-Konsole ändern, wenn die Objektgröße weniger als 160 GB beträgt. Bei größeren Objekten empfehlen wir, eine S3-Lebenszykluskonfiguration hinzuzufügen, um die Speicherklasse des Objekts zu ändern.
- Wenn Sie die Amazon S3 S3-Konsole verwenden, um die Speicherklasse für ein Objekt mit benutzerdefinierten Tags zu ändern, benötigen Sie die `s3:GetObjectTagging`

entsprechende Genehmigung. Wenn Sie die Speicherklasse für ein Objekt ändern, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Genehmigung. Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, wird die Speicherklasse für das Objekt aktualisiert, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

- Sie können Amazon S3 anweisen, die Speicherklasse von Objekten zu ändern, indem Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen. Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).
- Bei der Einrichtung der Replikation können Sie eine beliebige andere Speicherklasse für die replizierten Objekte einrichten. Sie können jedoch nicht in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeicherte Objekte replizieren. Weitere Informationen finden Sie unter [Replikationskonfiguration](#).

Einschränken von Zugriffsrichtlinienberechtigungen auf eine bestimmte Speicherklasse

Wenn Sie Zugriffsrichtlinienberechtigungen für Amazon-S3-Vorgänge erteilen, können Sie mit dem Bedingungsschlüssel `s3:x-amz-storage-class` einschränken, welche Speicherklasse beim Speichern hochgeladener Objekte verwendet werden soll. Wenn Sie beispielsweise die Berechtigung `s3:PutObject` erteilen, können Sie das Hochladen von Objekten auf eine bestimmte Speicherklasse einschränken. Eine Beispielrichtlinie finden Sie unter [Beispiel: Beschränkung von Objekt-Uploads auf Objekte mit einer bestimmten Speicherklasse](#).

Weitere Informationen zur Verwendung von Bedingungen in Richtlinien und die vollständige Liste der Bedingungsschlüssel in Amazon S3 finden Sie in den folgenden Themen:

- [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

- [Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln](#)

Langfristige Datenspeicherung mit S3 Glacier-Speicherklassen

Amazon S3 bietet mehrere S3 Glacier-Speicherklassen, die darauf ausgelegt sind, kostengünstige Lösungen für die Speicherung langfristiger Daten zu bieten, auf die nicht oft zugegriffen wird. Die S3 Glacier-Speicherklassen sind:

- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive

Sie wählen eine dieser Speicherklassen basierend darauf, wie oft Sie auf Ihre Daten zugreifen und wie schnell Sie sie abrufen müssen. Jede dieser Speicherklassen bietet dieselbe Haltbarkeit und Stabilität wie die Speicherkategorie S3 Standard, jedoch zu niedrigeren Speicherkosten. Weitere Informationen zu den S3 Glacier-Speicherklassen finden Sie unter <https://aws.amazon.com/s3/storage-classes/glacier/>.

Topics

- [Vergleich der S3 Glacier-Speicherklassen](#)
- [S3 Glacier Instant Retrieval](#)
- [S3 Glacier Flexible Retrieval](#)
- [S3 Glacier Deep Archive](#)
- [Archivierter Speicher](#)
- [Worin unterscheiden sich diese Speicherklassen vom S3 Glacier-Service](#)

Vergleich der S3 Glacier-Speicherklassen

Jede S3 Glacier-Speicherkategorie hat eine Mindestspeicherdauer für alle Objekte. Wenn Sie das Objekt vor Ablauf der Mindestspeicherkategorie löschen, überschreiben oder in eine andere Speicherkategorie verschieben, wird Ihnen die volle Mindestspeicherdauer in Rechnung gestellt.

Einige S3 Glacier-Speicherkategorien sind archivierbar, was bedeutet, dass die in diesen Kategorien gespeicherten Objekte archiviert werden und nicht für den Echtzeitzugriff verfügbar sind. Weitere Informationen finden Sie unter [Archivierter Speicher](#).

Speicherklassen, die für weniger häufige Zugriffsmuster mit längeren Abrufzeiten konzipiert sind, bieten niedrigere Speicherkosten. Preisinformationen finden Sie unter <https://aws.amazon.com/s3/pricing/>.

In der folgenden Tabelle sind die wichtigsten Punkte zusammengefasst, die bei der Auswahl einer S3 Glacier-Speicherklasse zu berücksichtigen sind:

S3 Glacier Instant Retrieval

Wir empfehlen die Verwendung von S3 Glacier Instant Retrieval für Langzeitdaten, auf die einmal pro Quartal zugegriffen wird und für die Abrufzeiten im Millisekundenbereich erforderlich sind. Diese Speicherklasse ist ideal für leistungskritische Anwendungsfälle wie das Hosten von Bildern, Filesharing-Anwendungen und das Speichern von Krankenakten für den Zugriff bei Terminen.

Die Speicherklasse S3 Glacier Instant Retrieval bietet Echtzeitzugriff auf Ihre Objekte mit derselben Latenz und Durchsatzleistung wie die Speicherklasse S3 Standard-IA. Im Vergleich zu S3 Standard-IA hat S3 Glacier Instant Retrieval niedrigere Speicherkosten, aber höhere Datenzugriffskosten.

Für Daten, die in der Speicherklasse S3 Glacier Instant Retrieval gespeichert sind, gilt eine minimale Objektgröße von 128 KB. Diese Speicherklasse hat außerdem eine Mindestspeicherdauer von 90 Tagen.

S3 Glacier Flexible Retrieval

Wir empfehlen die Verwendung von S3 Glacier Flexible Retrieval für Archivdaten, auf die ein- bis zweimal pro Jahr zugegriffen wird und für die kein sofortiger Zugriff erforderlich ist. S3 Glacier Flexible Retrieval bietet flexible Abrufzeiten, um Ihnen dabei zu helfen, die Kosten im Gleichgewicht zu halten. Die Zugriffszeiten reichen von wenigen Minuten bis hin zu Stunden und kostenlose Sammelabrufe. Diese Speicherklasse ist ideal für Backup und Disaster Recovery.

In S3 Glacier Flexible Retrieval gespeicherte Objekte werden archiviert und sind nicht für den Echtzeitzugriff verfügbar. Weitere Informationen finden Sie unter [Archivierter Speicher](#). Um auf diese Objekte zuzugreifen, initiieren Sie zunächst eine Wiederherstellungsanforderung, die eine temporäre Kopie des Objekts erstellt, auf die Sie zugreifen können, wenn die Anforderung abgeschlossen ist. Weitere Informationen finden Sie unter [Arbeiten mit archivierten Objekten](#). Wenn Sie ein Objekt wiederherstellen, können Sie eine Abrufstufe wählen, die Ihrem Anwendungsfall entspricht. Bei längeren Wiederherstellungszeiten sind die Kosten geringer.

Die folgenden Abrufstufen sind für S3 Glacier Flexible Retrieval verfügbar:

- **Beschleunigter Abruf** — Stellt das Objekt in der Regel innerhalb von 1—5 Minuten wieder her. Beschleunigte Abrufe unterliegen der Nachfrage. Um zuverlässige und vorhersehbare Wiederherstellungszeiten zu gewährleisten, empfehlen wir Ihnen, bereitgestellte Abrufkapazität zu erwerben. Weitere Informationen finden Sie unter [Bereitgestellte Kapazität](#).
- **Standardabruf** — Stellt das Objekt in der Regel innerhalb von 3—5 Stunden oder innerhalb von 1 Minute bis 5 Stunden wieder her, wenn Sie S3 Batch Operations verwenden. Weitere Informationen finden Sie unter [Objekte mit Batch Operations wiederherstellen](#).
- **Massenabruf** — Das Objekt wird in der Regel innerhalb von 5—12 Stunden wiederhergestellt. Massenabrufe sind kostenlos.

Die Mindestspeicherdauer für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval beträgt 90 Tage.

S3 Glacier Flexible Retrieval erfordert 40 KB an zusätzlichen Metadaten für jedes Objekt. Dazu gehören 32 KB an Metadaten, die für die Identifizierung und den Abruf Ihrer Daten erforderlich sind. Diese werden zum Standardtarif für S3 Glacier Flexible Retrieval berechnet. Weitere 8 KB Daten sind erforderlich, um den benutzerdefinierten Namen und die Metadaten für archivierte Objekte zu verwalten, und werden zum S3-Standardtarif berechnet.

S3 Glacier Deep Archive

Wir empfehlen die Verwendung von S3 Glacier Deep Archive für Archivdaten, auf die weniger als einmal pro Jahr zugegriffen wird. Diese Speicherklasse ist für die Aufbewahrung von Datensätzen über mehrere Jahre konzipiert, um Compliance-Anforderungen zu erfüllen. Sie kann auch für Backup oder Disaster Recovery oder für Daten verwendet werden, auf die Sie selten zugreifen und auf deren Abruf Sie bis zu 72 Stunden warten können. S3 Glacier Deep Archive ist die kostengünstigste Speicheroption in AWS.

In S3 Glacier Deep Archive gespeicherte Objekte sind archiviert und für den Echtzeitzugriff nicht verfügbar. Weitere Informationen finden Sie unter [Archivierter Speicher](#). Um auf diese Objekte zuzugreifen, initiieren Sie zunächst eine Wiederherstellungsanforderung, die eine temporäre Kopie des Objekts erstellt, auf die Sie zugreifen können, wenn die Anfrage abgeschlossen ist. Weitere Informationen finden Sie unter [Arbeiten mit archivierten Objekten](#). Wenn Sie ein Objekt wiederherstellen, können Sie eine Abrufstufe wählen, die Ihrem Anwendungsfall entspricht. Bei längeren Wiederherstellungszeiten sind die Kosten geringer.

Die folgenden Abrufstufen sind für S3 Glacier Deep Archive verfügbar:

- Standardabruf — Stellt das Objekt in der Regel innerhalb von 12 Stunden oder innerhalb von 9—12 Stunden wieder her, wenn Sie S3 Batch Operations verwenden. Weitere Informationen finden Sie unter [Objekte mit Batch Operations wiederherstellen](#).
- Massenabruf — Stellt das Objekt in der Regel innerhalb von 48 Stunden wieder her, und das zu einem Bruchteil der Kosten des Standard-Abrufs.

Die Mindestspeicherdauer für Objekte in der Speicherklasse S3 Glacier Deep Archive beträgt 180 Tage.

S3 Glacier Deep Archive benötigt 40 KB an zusätzlichen Metadaten für jedes Objekt. Dazu gehören 32 KB an Metadaten, die zur Identifizierung und zum Abrufen Ihrer Daten erforderlich sind. Diese werden zum Standardtarif für S3 Glacier Deep Archive berechnet. Weitere 8 KB Daten sind erforderlich, um den benutzerdefinierten Namen und die Metadaten für archivierte Objekte beizubehalten, und werden zum S3-Standardtarif berechnet.

Archivierter Speicher

S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind Archivspeicherklassen. Das heißt, wenn Sie ein Objekt in diesen Speicherklassen speichern, wird dieses Objekt archiviert und es kann nicht direkt darauf zugegriffen werden. Um auf ein archiviertes Objekt zuzugreifen, senden Sie eine Wiederherstellungsanforderung für das Objekt und warten dann, bis der Dienst das Objekt wiederherstellt. Die Wiederherstellungsanforderung stellt eine temporäre Kopie des Objekts wieder her. Diese Kopie wird gelöscht, wenn die von Ihnen in der Anforderung angegebene Dauer abgelaufen ist. Weitere Informationen finden Sie unter [Arbeiten mit archivierten Objekten](#).

Für diese Speicherklassen sind 40 KB an zusätzlichen Metadaten für jedes archivierte Objekt erforderlich. Dazu gehören 32 KB an Metadaten, die zum Identifizieren und Abrufen Ihrer Daten erforderlich sind. Diese werden zum Standardtarif für diese Speicherklasse berechnet. Weitere 8 KB Daten sind erforderlich, um den benutzerdefinierten Namen und die Metadaten für archivierte Objekte beizubehalten, und werden zum S3-Standardtarif berechnet.

Objekte in diesen Speicherklassen werden zu den S3-Standardtarifen für Speicherklassen in Rechnung gestellt, wenn Sie sie mithilfe von mehrteiligen Uploads hochladen. Weitere Informationen finden Sie unter [Mehrteiliger Upload und Preise](#).

Sie können archivierte Objekte in diesen Speicherklassen mit bis zu 1.000 Transaktionen pro Sekunde (TPS) an [Objektwiederherstellungsanforderungen pro Konto pro Konto wiederherstellen](#).
AWS-Region

Worin unterscheiden sich diese Speicherklassen vom S3 Glacier-Service

Die S3 Glacier-Speicherklassen sind Teil des Amazon S3 S3-Service und speichern Daten als Objekte in S3-Buckets. Sie können Objekte in diesen Speicherklassen mithilfe der S3-Konsole oder programmgesteuert mithilfe der S3-APIs oder SDKs verwalten. Wenn Sie Objekte in S3 Glacier-Speicherklassen speichern, können Sie S3-Funktionen wie erweiterte Verschlüsselung, Objekt-Tagging und S3 Lifecycle-Konfigurationen verwenden, um den Datenzugriff und die Kosten zu verwalten.

Important

Wir empfehlen, die S3 Glacier-Speicherklassen innerhalb des Amazon S3 S3-Service für all Ihre langfristigen Daten zu verwenden.

Der Amazon S3 Glacier (S3 Glacier) Service ist ein separater Service, der Daten als Archive in Tresoren speichert. Dieser Service unterstützt keine Amazon S3 S3-Funktionen und bietet keine Konsolenunterstützung für das Hoch- und Herunterladen von Daten. Wir empfehlen nicht, den S3 Glacier-Service für Ihre langfristigen Daten zu verwenden. Auf die in diesem Service gespeicherten Daten kann über den Amazon S3 S3-Service nicht zugegriffen werden. Informationen zum S3 Glacier-Service finden Sie im [Amazon S3 Glacier Developer Guide](#). Informationen zum Übertragen von Daten vom Amazon S3 Glacier-Service zu einer Speicherkategorie in Amazon S3 finden Sie unter [Datenübertragung von Amazon S3 Glacier Vaults zu Amazon S3](#) in der AWS Lösungsbibliothek.

Amazon S3 Intelligent Tiering

Die Speicherkategorie S3 Intelligent-Tiering wurde entwickelt, um die Speicherkosten zu optimieren, indem Daten automatisch auf die kostengünstigste Zugriffsebene verschoben werden, wenn sich die Zugriffsmuster ändern, ohne dass Betriebsaufwand oder Leistungseinbußen auftreten. Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt die Objekte, auf die nicht zugegriffen wurde, automatisch in kostengünstigere Zugriffsebenen.

S3 Intelligent-Tiering ermöglicht automatische Speicherkosteneinsparungen in drei Zugriffsebenen mit niedriger Latenz und hohem Durchsatz. Für Daten, auf die asynchron zugegriffen werden kann, können Sie innerhalb der S3 Intelligent-Tiering-Speicherkategorie die automatische Archivfunktionen aktivieren. Bei S3 Intelligent-Tiering werden keine Abrufgebühren erhoben. Wenn später auf ein Objekt in der Stufe Infrequent Access oder Archive Instant Access zugegriffen wird, wird es

automatisch zurück in die Stufe Frequent Access verschoben. Es werden keine Zusatzgebühren erhoben, wenn Objekte innerhalb der S3-Intelligent-Tiering-Speicherklasse zwischen Zugriffsstufen verschoben werden.

S3 Intelligent-Tiering ist die empfohlene Speicherklasse für Daten mit unbekanntem, sich ändernden oder unvorhersehbarem Zugriffsmustern, unabhängig von der Objektgröße oder dem Aufbewahrungszeitraum, wie Data Lakes, Datenanalytik und neue Anwendungen.

Weitere Informationen zur Verwendung von S3 Intelligent-Tiering finden Sie in den folgenden Abschnitten:

Themen

- [So funktioniert S3 Intelligent-Tiering](#)
- [Verwenden von S3 Intelligent-Tiering](#)
- [Verwenden von S3 Intelligent-Tiering](#)

So funktioniert S3 Intelligent-Tiering

Die Speicherklasse Amazon S3 Intelligent-Tiering speichert Objekte automatisch in drei Zugriffsebenen. Eine Stufe ist für häufigen Zugriff optimiert, eine kostengünstigere Stufe ist für seltenen Zugriff optimiert und eine andere sehr kostengünstige Stufe ist für selten zugängliche Daten optimiert. Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt Objekte automatisch in die Ebene für seltenen Zugriff, wenn auf sie 30 aufeinanderfolgende Tage lang nicht zugegriffen wurde. Nach 90 Tagen ohne Zugriff werden die Objekte ohne Leistungseinbußen oder Betriebsaufwand auf die Ebene des Archivs Instant Access verschoben.

Aktivieren Sie Archivierungsfunktionen, um zwei zusätzliche Zugriffsebenen hinzuzufügen und die niedrigsten Speicherkosten für Daten zu erhalten, auf die in Minuten bis Stunden zugegriffen werden kann. Sie können Objekte schichtweise auf die Archive Access Tier oder Deep Archive Access Tier oder beiden reduzieren. Mit Archive Access verschiebt S3 Intelligent-Tiering Objekte, auf die mindestens 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde, in die Ebene „Archive Access“. Mit Deep Archive Access verschiebt S3 Intelligent-Tiering Objekte, auf die mindestens 180 aufeinanderfolgende Tage lang nicht zugegriffen wurde, in die Ebene „Deep Archive Access“. Für beide Ebenen können Sie die Anzahl der Tage ohne Zugriff je nach Bedarf konfigurieren.

Die folgenden Aktionen stellen einen Zugriff dar, der ein Herabstufen Ihrer Objekte in die Ebene „Archive Access“ oder „Deep Archive Access“ verhindert:

- Herunterladen oder Kopieren eines Objekts über die Amazon S3-Konsole.
- Aufrufen von [CopyObject](#), [UploadPartCopy](#) oder Replizieren bestehender Objekte mit S3- Batch-Replikation. In diesen Fällen werden die Quellobjekte der Kopier- oder Replikationsvorgänge hochgestuft.
- Aufrufen von [GetObject](#), [PutObject](#), [RestoreObject](#), [CompleteMultipartUpload](#), [ListParts](#) oder [SelectObjectContent](#).

Wenn beispielsweise vor Ablauf der von Ihnen festgelegte Anzahl von Tagen ohne Zugriff (z. B. 180 Tage) über `SelectObjectContent` ein Zugriff auf Ihre Objekte erfolgt, wird durch diese Aktion der Timer zurückgesetzt. Ihre Objekte werden erst in die Ebene „Archive Access“ oder „Deep Archive Access“ verschoben, wenn die Zeit nach der letzten `SelectObjectContent`-Anforderung die von Ihnen festgelegte Anzahl von Tagen erreicht hat.

Wenn später auf ein Objekt in der Stufe Infrequent Access oder Archive Instant Access zugegriffen wird, wird es automatisch zurück in die Stufe Frequent Access verschoben.

Die folgenden Aktionen stellen einen Zugriff dar, bei dem Objekte automatisch von der Ebene „Infrequent Access“ oder „Archive Instant Access“ zurück in die Ebene „Frequent Access“ verschoben werden:

- Herunterladen oder Kopieren eines Objekts über die Amazon S3-Konsole.
- Aufrufen von [CopyObject](#), [UploadPartCopy](#) oder Replizieren bestehender Objekte mit Batch-Replikation. In diesen Fällen werden die Quellobjekte der Kopier- oder Replikationsvorgänge hochgestuft.
- Aufrufen von [GetObject](#), [PutObject](#), [RestoreObject](#), [CompleteMultipartUpload](#) oder [ListParts](#).

Andere Aktionen stellen keinen Zugriff dar, bei dem Objekte automatisch von der Ebene „Infrequent Access“ oder „Archive Instant Access“ zurück in die Ebene „Frequent Access“ verschoben werden. Im Folgenden finden Sie ein Beispiel, keine endgültige Liste, solcher Aktionen:

- Aufrufen von [HeadObject](#), [GetObjectTagging](#), [PutObjectTagging](#), [ListObjects](#), [ListObjectsV2](#) oder [ListObjectVersions](#).
- Das Aufrufen von [SelectObjectContent](#) stellt keinen Zugriff dar, bei dem Objekte zur Ebene „Frequent Access“ hochgestuft werden. Darüber hinaus wird dadurch nicht verhindert, dass Objekte von der Stufe „Frequent Access“ auf die Stufe „Infrequent Access“ und von dort aus auf die Stufe „Archive Instant Access“ herabgestuft werden.

Sie können S3 Intelligent-Tiering als Standardspeicherklasse für neu erstellte Daten konfigurieren, indem Sie INTELLIGENT-TIERING in Ihrem [PutBucketIntelligentTieringConfiguration](#)-Anforderungs-Header festlegen. S3 Intelligent-Tiering ist auf 99,9 % Verfügbarkeit und 99,9999999 % Haltbarkeit ausgelegt.

Note

Wenn die Größe eines Objekts weniger als 128 KB beträgt, wird es nicht überwacht und ist nicht für das automatische Tiering geeignet. Kleinere Objekte werden immer in der Stufe für häufige Zugriffe gespeichert.

S3-Intelligent-Tiering-Zugriffsebenen

Im folgenden Abschnitt werden die verschiedenen automatischen und optionalen Zugriffsebenen erläutert. Wenn Objekte zwischen Zugriffsebenen verschoben werden, bleibt die Speicherklasse gleich (S3 Intelligent-Tiering).

Stufe für häufige Zugriffe (automatisch)

Hierbei handelt es sich um die Standardzugriffsebene, in der Objekte, die in S3 Intelligent-Tiering erstellt oder überführt sind, seinen Lebenszyklus beginnt. Ein Objekt verbleibt in dieser Ebene, solange darauf zugegriffen wird. Die Stufe für häufige Zugriffe bietet niedrige Latenz und hohe Durchsatzleistung.

Stufe für seltene Zugriffe (automatisch)

Wenn 30 aufeinanderfolgende Tage auf ein Objekt nicht zugegriffen wird, wird das Objekt in die Stufe für seltene Zugriffe verschoben. Die Stufe für seltene Zugriffe bietet niedrige Latenz und hohe Durchsatzleistung.

Stufe Archive Instant Access (automatisch)


Wenn 90 aufeinanderfolgende Tage auf ein Objekt nicht zugegriffen wird, wird das Objekt in die Stufe Archive Instant Access verschoben. Die Stufe Archive Instant Access bietet niedrige Latenzzeiten und hohe Durchsatzleistung.

Archivzugriffsebene (optional)

S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Archivzugriffsebene für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Archive

Access automatisch Objekte, auf die mindestens 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde. Sie können die letzte Zugriffszeit für die Archivierung auf maximal 730 Tage verlängern. Die Archivzugriffsebene hat dieselbe Leistung wie die Speicherklasse [S3 Glacier Flexible Retrieval](#).

Die Standardabrufzeiten für diese Zugriffsebene können zwischen 3 und 5 Stunden liegen. Wenn Sie Ihre Wiederherstellungsanfrage mithilfe von S3 Batch Operations starten, beginnt Ihre Wiederherstellung innerhalb von Minuten. Weitere Informationen zu den Abrufoptionen finden -Zeiten finden Sie unter [the section called “Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access”](#).


 Note

Aktivieren Sie die Stufe Archive Access nur für 90 Tage, wenn Sie die Stufe Instant Access Archiv umgehen möchten. Die Ebene „Archive Access“ liefert etwas niedrigere Speicherkosten mit Abrufzeiten von Minuten zu Stunde. Die Stufe Archive Instant Access bietet Millisekunden-Zugriff und hohe Durchsatzleistung.

Zugriffsebene für Deep Archive (optional)

S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Stufe Deep Archive Access für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Deep Archive Access automatisch Objekte, auf die mindestens 180 aufeinanderfolgende Tage lang nicht zugegriffen wurde. Sie können die letzte Zugriffszeit für die Archivierung auf maximal 730 Tage verlängern. Die Deep-Archive-Zugriffsebene hat dieselbe Leistung wie die [S3 Glacier Deep Archive](#)-Speicherklasse.

Der Standardabruf von Objekten in dieser Zugriffsebene erfolgt innerhalb von 12 Stunden. Wenn Sie Ihre Wiederherstellungsanfrage mithilfe von S3 Batch Operations starten, beginnt Ihre Wiederherstellung innerhalb von 9 Stunden. Weitere Informationen zu den Abrufoptionen finden -Zeiten finden Sie unter [the section called “Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access”](#).

 Note

Aktivieren Sie die Ebenen Archivzugriff und Deep Archive Access nur, wenn Ihre Anwendung asynchron auf Ihre Objekte zugreifen kann. Wenn das Objekt, das Sie abrufen, in den

Ebenen „Archive Access“ oder „Deep Archive Access“ gespeichert ist, müssen Sie das Objekt zunächst mithilfe des `RestoreObject`-Vorgangs wieder herstellen.

Verwenden von S3 Intelligent-Tiering

Sie können die Speicherklasse S3 Intelligent-Tiering verwenden, um die Speicherkosten automatisch zu optimieren. S3 Intelligent-Tiering ermöglicht automatische Kosteneinsparungen durch Verschieben von Daten auf granularer Objektebene zwischen den Zugriffsebenen, wenn sich die Zugriffsmuster ändern. Für Daten, auf die asynchron zugegriffen werden kann, können Sie wählen, ob Sie die automatische Archivierung innerhalb der Speicherklasse S3 Intelligent-Tiering mithilfe von AWS Management Console AWS CLI, oder Amazon S3 aktivieren möchten. API

Verschieben von Daten nach S3 Intelligent-Tiering

Es gibt zwei Möglichkeiten, Daten in S3 Intelligent-Tiering zu übertragen. Sie können [PUT](#)Daten direkt in S3 Intelligent-Tiering übertragen, indem Sie sie `INTELLIGENT_TIERING` im `x-amz-storage-class` Header angeben oder S3 Lifecycle-Konfigurationen so konfigurieren, dass Objekte von S3 Standard oder S3 Standard-Infrequent Access auf S3 Intelligent-Tiering umgestellt werden.

Daten mithilfe von Direct auf S3 Intelligent-Tiering hochladen PUT

Wenn Sie mithilfe der [PUT](#)APIOperation ein Objekt in die S3 Intelligent-Tiering-Speicherklasse hochladen, geben Sie S3 Intelligent-Tiering im Anforderungsheader an. [x-amz-storage-class](#)

Die folgende Anforderung speichert das Image, `my-image.jpg`, im `myBucket`-Bucket. Die Anforderung verwendet den `x-amz-storage-class`-Header, um anzufordern, dass das Objekt mit der Speicherklasse S3 Intelligent-Tiering gespeichert ist.

Example

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.<Region>.amazonaws.com (http://amazonaws.com/)
Date: Wed, 1 Sep 2021 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: INTELLIGENT_TIERING
```

Übergang von Daten zu S3 Intelligent-Tiering von S3 Standard oder S3 Standard-Infrequent Access mit S3-Lebenszyklus

Sie können Regeln zu Ihrer S3-Lebenszykluskonfiguration hinzufügen und damit Amazon S3 anweisen, Objekte von einer Speicherklasse in eine andere zu wechseln. Informationen zu unterstützten Übergängen und zugehörigen Einschränkungen finden Sie unter [Migrieren von Objekten mit S3-Lebenszyklus](#).

Sie können S3-Lebenszykluskonfigurationen auf der Bucket- oder Präfixebene angeben. In dieser S3-Lebenszyklus-Konfigurationsregel spezifiziert der Filter ein Schlüsselpräfix (documents/). Aus diesem Grund gilt die Regel für Objekte mit dem Schlüsselnamenpräfix documents/, wie beispielsweise documents/doc1.txt und documents/doc2.txt. Die Regel gibt eine Transition-Aktion an, die Amazon S3 anweist, Objekte 0 Tage nach der Erstellung in die Speicherklasse S3 Intelligent-Tiering zu übertragen. In diesem Fall können Objekte um Mitternacht nach ihrer Erstellung auf S3 Intelligent-Tiering umgestellt werden. UTC

Example

```
<LifecycleConfiguration>
  <Rule>
    <ID>ExampleRule</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>INTELLIGENT_TIERING</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Ein versionsfähiger Bucket behält eine aktuelle Objektversion sowie und null oder mehr langfristige Objektversionen bei. Sie können separate Lebenszyklusrichtlinie für aktuelle und nicht aktuelle Objektversionen definieren.

Weitere Informationen finden Sie unter [Elemente der Lebenszyklus-Konfiguration](#).

Aktivieren der Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access

Um die niedrigsten Speicherkosten für Daten zu erzielen, auf die innerhalb von Minuten bis Stunden zugegriffen werden kann, können Sie eine oder beide Archivzugriffsebenen aktivieren, indem Sie mit Amazon S3 eine Konfiguration auf Bucket-, Präfix- oder Objekt-Tag-Ebene erstellen. API, AWS Management Console, AWS CLI

Verwenden der S3-Konsole

So aktivieren Sie die automatische S3 Intelligent-Tiering-Archivierung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
3. Wählen Sie Properties (Eigenschaften).
4. Navigieren Sie zu den S3 Intelligent-Tiering-Archive-Konfigurationen und wählen Sie Erstellen einer Konfiguration aus.
5. Geben Sie im Abschnitt Archiveinstellungen einen beschreibenden Konfigurationsnamen für die Konfiguration des S3 Intelligent-Tiering-Archivs an.
6. Wählen Sie unter Auswählen eines Konfigurationsbereichs einen zu verwendenden Konfigurationsbereich aus. Optional können Sie den Konfigurationsbereich auf bestimmte Objekte innerhalb eines Buckets beschränken, indem Sie ein gemeinsames Präfix, ein Objekt-Tag oder eine Kombination dieser beiden verwenden.
 - a. Um den Umfang der Konfiguration einzuschränken, wählen Sie Begrenzen des Umfangs dieser Konfiguration mithilfe eines oder mehrerer Filter aus.
 - b. Um den Bereich der Konfiguration mit einem einzigen Präfix zu beschränken, geben Sie den Präfix unter Präfix ein.
 - c. Um den Umfang der Konfiguration mithilfe von Objekt-Tags einzuschränken, wählen Sie Hinzufügen eines Tags, und geben Sie einen Wert für Schlüssel ein.
7. Wählen Sie unter Status Aktivieren aus.
8. Wählen Sie im Abschnitt Archiveinstellungen eine oder beide der Archivzugriffsebenen aus, die aktiviert werden sollen.
9. Wählen Sie Create (Erstellen) aus.

Verwenden Sie den AWS CLI

Sie können die folgenden AWS CLI Befehle verwenden, um S3 Intelligent-Tiering-Konfigurationen zu verwalten:

- [delete-bucket-intelligent-tiering-configuration](#)
- [get-bucket-intelligent-tiering-configuration](#)
- [list-bucket-intelligent-tiering-configurations](#)
- [put-bucket-intelligent-tiering-configuration](#)

Anweisungen zur Einrichtung von finden Sie unter AWS CLI. [Entwickeln mit Amazon S3 über die AWS CLI](#)

Wenn Sie den verwenden AWS CLI, können Sie die Konfiguration nicht als XML Datei angeben. Sie müssen JSON stattdessen die angeben. Im Folgenden finden Sie ein Beispiel für eine XML S3 Intelligent-Tiering-Konfiguration und eine entsprechende KonfigurationJSON, die Sie in einem Befehl angeben können. AWS CLI

Im folgenden Beispiel wird eine S3-Intelligent-Tiering-Konfiguration in den angegebenen Bucket übertragen.

Example [put-bucket-intelligent-tiering-configuration](#)

JSON

```
{
  "Id": "string",
  "Filter": {
    "Prefix": "string",
    "Tag": {
      "Key": "string",
      "Value": "string"
    },
    "And": {
      "Prefix": "string",
      "Tags": [
        {
          "Key": "string",
          "Value": "string"
        }
      ]
    }
  }
}
```

```

    ...
  ]
}
},
"Status": "Enabled"|"Disabled",
"Tierings": [
  {
    "Days": integer,
    "AccessTier": "ARCHIVE_ACCESS"|"DEEP_ARCHIVE_ACCESS"
  }
  ...
]
}

```

XML

```

PUT /?intelligent-tiering&id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<IntelligentTieringConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>string</Id>
  <Filter>
    <And>
      <Prefix>string</Prefix>
      <Tag>
        <Key>string</Key>
        <Value>string</Value>
      </Tag>
      ...
    </And>
    <Prefix>string</Prefix>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </Filter>
  <Status>string</Status>
  <Tiering>
    <AccessTier>string</AccessTier>
    <Days>integer</Days>
  </Tiering>
  ...
</IntelligentTieringConfiguration>

```


Verwenden der Operation PUT API

Sie können den [PutBucketIntelligentTieringConfiguration](#)-Vorgang für einen angegebenen Bucket und bis zu 1000 S3-Intelligent-Tiering-Konfigurationen pro Bucket nutzen. Mit einem gemeinsamen Präfix oder Objekt-Tag können Sie festlegen, welche Objekte innerhalb eines Buckets für die Archivzugriffsstufen berechtigt sind. Durch die Verwendung eines gemeinsamen Präfixes oder Objekt-Tags können Sie auf bestimmte Geschäftsanwendungen, Workflows oder interne Organisationen ausrichten. Außerdem haben Sie die Flexibilität, die Stufe Archivzugriff, die Stufe Deep Archive Access oder beide zu aktivieren.

Erste Schritte mit S3 Intelligent-Tiering

Weitere Informationen zur Verwendung von S3 Intelligent-Tiering finden Sie unter [Tutorial: Erste Schritte mit S3 Intelligent-Tiering](#).

Verwenden von S3 Intelligent-Tiering

Die Speicherklasse S3 Intelligent-Tiering ermöglicht automatische Speicherkosteneinsparungen in drei Zugriffsebenen mit niedriger Latenz und hohem Durchsatz. Sie bietet außerdem optionale Archivierungsfunktionen, mit denen Sie die niedrigsten Speicherkosten in der Cloud für Daten erzielen, auf die innerhalb von Minuten bis Stunden zugegriffen werden kann. Die Speicherklasse S3 Intelligent-Tiering unterstützt alle Amazon-S3-Funktionen, einschließlich der folgenden:

- S3 Inventory zur Überprüfung der Zugriffsebene von Objekten
- S3-Replikation, für die Replikation von Daten auf beliebige AWS-Region
- S3 Storage Lens zum Anzeigen von Speichernutzungs- und Aktivitätsmetriken
- Serverseitige Verschlüsselung, für den Schutz von Objektdaten
- S3-Objektsperre, um versehentliches Löschen von Daten zu verhindern
- AWS PrivateLink, um über einen privaten Endpunkt in einer Virtual Private Cloud (VPC) auf Amazon S3 zuzugreifen

Festlegen welche der S3-Intelligent-Tiering-Zugriffsebene-Objekte gespeichert sind in

Um eine Liste Ihrer Objekte und der entsprechenden Metadaten, einschließlich ihrer S3 Intelligent-Tiering-Zugriffsebene, zu erhalten, können Sie Folgendes verwenden. [the section called “Verwalten des Bestands”](#) S3 Inventory bietet CSV, ORC oder Parquet-Ausgabedateien, die Ihre Objekte und

die entsprechenden Metadaten auflisten. Sie können diese Bestandsberichte entweder täglich oder wöchentlich für einen Amazon-S3-Bucket oder ein gemeinsames Präfix erhalten. (Gemeinsames Präfix bezieht sich auf Objekte, die Namen haben, die mit einer bestimmten Zeichenfolge beginnen.)

Anzeigen des Archivstatus eines Objekts in S3 Intelligent-Tiering

Wenn Sie eine Benachrichtigung erhalten möchten, sobald ein Objekt innerhalb der Speicherklasse S3 Intelligent-Tiering entweder auf die Stufe Archive Access oder die Stufe Deep Archive Access verschoben wurde, können Sie S3-Ereignisbenachrichtigungen einrichten. Weitere Informationen finden Sie unter [Aktivieren von Ereignisbenachrichtigungen](#).

Amazon S3 kann Ereignisbenachrichtigungen in einem Amazon Simple Notification Service (Amazon SNS)-Thema, einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange oder einer AWS Lambda -Funktion veröffentlichen. Weitere Informationen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Das Folgende ist ein Beispiel für eine Nachricht, die Amazon S3 sendet, um ein `s3: IntelligentTiering`-Ereignis zu veröffentlichen. Weitere Informationen finden Sie unter [the section called "Struktur von Ereignismeldungen"](#).

```
{
  "Records": [
    {
      "eventVersion": "2.3",
      "eventSource": "aws:s3",
      "awsRegion": "us-west-2",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "IntelligentTiering",
      "userIdentity": {
        "principalId": "s3.amazonaws.com"
      },
      "requestParameters": {
        "sourceIPAddress": "s3.amazonaws.com"
      },
      "responseElements": {
        "x-amz-request-id": "C3D13FE58DE4C810",
        "x-amz-id-2": "FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvAN0jpD"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",

```

```
    "bucket":{
      "name":"mybucket",
      "ownerIdentity":{
        "principalId":"A3NL1K0ZZKExample"
      },
      "arn":"arn:aws:s3:::mybucket"
    },
    "object":{
      "key":"HappyFace.jpg",
      "size":1024,
      "eTag":"d41d8cd98f00b204e9800998ecf8427e",
    }
  },
  "intelligentTieringEventData":{
    "destinationAccessTier": "ARCHIVE_ACCESS"
  }
}
]
```

Sie können auch eine [HEAD-Objekt-Anfrage](#) nutzen, um den Archivstatus eines Objekts anzuzeigen. Wenn ein Objekt in der Speicherklasse S3 Intelligent-Tiering gespeichert wird und sich in einer der Archivebenen befindet, zeigt die HEAD-Objektantwort die aktuelle Archivebene an. Um die Archivstufe anzuzeigen, verwendet die Anfrage den [x-amz-archive-status](#)-Header an.

Die folgende HEAD-Objektanforderung gibt die Metadaten eines Objekts zurück (in diesem Fall *my-image.jpg*).

Example

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.region.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

Darüber hinaus können Sie HEAD-Objektanforderungen verwenden, um den Status einer `restore-object`-Anforderung zu überwachen. Wenn die Archivwiederherstellung ausgeführt wird, enthält die HEAD-Objekt-Antwort den [x-amz-restore](#)-Header.

Das folgende Beispiel veranschaulicht die HEAD-Objektantwort, die ein mit S3 Intelligent-Tiering archiviertes Objekt mit einer Wiederherstellungsanforderung anzeigt.

Example

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp3Izs1NnwBZeu7M19iI8UbxMbi0A8AirHANJBo+hEftBuiESACOMJp
x-amz-request-id: E5CEFCB143EB505A
Date: Fri, 13 Nov 2020 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
ETag: "1accb31fcf202eba0c0f41fa2f09b4d7"
x-amz-storage-class: 'INTELLIGENT_TIERING'
x-amz-archive-status: 'ARCHIVE_ACCESS'
x-amz-restore: 'ongoing-request="true"'
x-amz-restore-request-date: 'Fri, 13 Nov 2020 00:20:00 GMT'
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access

Um auf Objekte in den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access zuzugreifen, müssen Sie eine [Wiederherstellungsanforderung](#) initiieren und dann warten, bis das Objekt in die Stufe für häufigen Zugriff verschoben wird. Weitere Informationen zu archivierten Objekten finden Sie unter [the section called "Arbeiten mit archivierten Objekten"](#)

Wenn Sie ein Objekt aus den Stufen Archive Access oder Deep Archive Access wiederherstellen, wird das Objekt zurück in die Stufe für häufige Zugriffe übergehen. Wenn anschließend nach 30 aufeinanderfolgenden Tagen nicht auf das Objekt zugegriffen wird, wird es automatisch in die Stufe „Infrequent Access“ verschoben. Das Objekt wechselt dann nach mindestens 90 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die Stufe „Archive Access“. Das Objekt wechselt nach mindestens 180 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die Stufe „Deep Archive Access“. Weitere Informationen finden Sie unter [the section called "So funktioniert S3 Intelligent-Tiering"](#).

Sie können ein archiviertes Objekt mithilfe der Amazon S3 S3-Konsole, S3 Batch Operations, der Amazon S3 S3-REST-API, der AWS SDKs oder der AWS Command Line Interface (AWS CLI) wiederherstellen. Weitere Informationen finden Sie unter [the section called "Arbeiten mit archivierten Objekten"](#).

Verwaltung des Lebenszyklus von Objekten

Um Ihre Objekte so zu verwalten, dass sie während ihres gesamten Lebenszyklus kostengünstig gespeichert werden, erstellen Sie eine Amazon S3 S3-Lifecycle-Konfiguration. Eine Amazon S3-Lifecycle-Konfiguration besteht aus einer Reihe von Regeln, die Aktionen definieren, die Amazon S3 auf eine Gruppe von Objekten anwendet. Es gibt zwei Aktionstypen:

- **Übergangsktionen** – Diese Aktionen definieren, wann Objekte in eine andere Speicherklasse übergehen. Beispielsweise können Sie festlegen, dass Objekte 30 Tage nach ihrer Erstellung in die Speicherklasse S3 Standard-IA (IA steht für „Infrequent Access“, seltener Zugriff) übergehen und ein Jahr nach ihrer Erstellung in der Speicherklasse S3 Glacier Flexible Retrieval archiviert werden sollen. Weitere Informationen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Mit Lebenszyklus-Überführungsanforderungen sind Kosten verbunden. Informationen zu den Preisen finden Sie unter [Amazon S3 – Preise](#).

- **Ablaufaktionen** – Diese Aktionen definieren, wann Objekte ablaufen. In Amazon S3 werden die abgelaufenen Objekte für Sie gelöscht.

Mit dem Ablauf des Lebenszyklus sind nur dann potenzielle Kosten verbunden, wenn Sie Objekte in einer Speicherklasse mit einer Mindestspeicherdauer ablaufen lassen. Weitere Informationen finden Sie unter [Gebühren für Mindestspeicherdauer](#).

Important

Sie können keine Bucket-Richtlinie verwenden, um Löschungen oder Übergänge durch eine S3-Lebenszyklusregel zu verhindern. Selbst wenn Ihre Bucket-Richtlinie beispielsweise alle Aktionen für alle Principals ablehnt, funktioniert Ihre S3-Lifecycle-Konfiguration weiterhin wie gewohnt.

Bestehende und neue Objekte

Wenn Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie beispielsweise heute eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte 30 Tage nach ihrer Erstellung ablaufen, stellt Amazon S3 eine Warteschlange für die Entfernung aller vorhandenen Objekte, die älter als 30 Tage sind, in die Warteschlange.

Änderungen bei der Abrechnung

Wenn es zwischen dem Zeitpunkt, an dem ein Objekt für eine Lebenszyklusaktion berechtigt wird, und dem Zeitpunkt, zu dem Amazon S3 Ihr Objekt überträgt oder abläuft, zu Verzögerungen kommt, werden die Abrechnungsänderungen angewendet, sobald das Objekt für die Lebenszyklusaktion berechtigt ist. Wenn ein Objekt beispielsweise abläuft und Amazon S3 das Objekt nicht sofort ablaufen lässt, wird Ihnen nach Ablauf der Ablaufzeit kein Speicherplatz in Rechnung gestellt.

Die einzige Ausnahme von diesem Verhalten ist, wenn Sie eine Lebenszyklusregel für den Übergang zur S3-Intelligent-Tiering-Speicherklasse haben. In diesem Fall treten Abrechnungsänderungen erst auf, wenn das Objekt in S3 Intelligent-Tiering übergeben worden ist.

Weitere Informationen zu S3-Lebenszyklusregeln finden Sie unter [Elemente der Lebenszyklus-Konfiguration](#).

Überwachung der Auswirkungen von Lebenszyklusregeln

Informationen zur Überwachung der Auswirkungen von Aktualisierungen, die durch aktive Lebenszyklusregeln vorgenommen wurden, finden Sie unter [the section called “Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?”](#).

Verwaltung des gesamten Lebenszyklus von Objekten

Mit den S3 Lifecycle-Konfigurationsregeln können Sie Amazon S3 anweisen, Objekte in kostengünstigere Speicherklassen zu überführen, zu archivieren oder zu löschen. z. B.:

- Wenn Sie regelmäßig Protokolle in einen Bucket hochladen, werden diese möglicherweise nur für eine Woche oder einen Monat von Ihrer Anwendung benötigt. Anschließend sollen diese gelöscht werden.
- Auf einige Dokumente erfolgt für einen begrenzten Zeitraum ein häufiger Zugriff. Danach wird nur selten auf diese zugegriffen. Irgendwann benötigen Sie keinen Echtzeitzugriff mehr auf die Dokumente, aber seitens Ihrer Organisation oder der Gesetzgebung müssen Sie diese noch für einen gewissen Zeitraum archivieren. Nach Ablauf dieser Zeitspanne können Sie sie löschen.
- Sie können auch einige Datentypen primär für Archivierungszwecke in Amazon S3 hochladen. Sie könnten beispielsweise digitale Medien, Finanz- und Gesundheitsdaten, rohe Genomsequenzdaten, langfristige Datenbank-Backups und Daten archivieren, die aus Gründen der Einhaltung gesetzlicher Vorschriften aufbewahrt werden müssen.

Indem Sie S3-Lifecycle-Aktionen kombinieren, um den gesamten Lebenszyklus eines Objekts zu verwalten. Angenommen, die von Ihnen erstellten Objekte haben einen definierten Lebenszyklus. Anfänglich erfolgt ein häufiger Zugriff auf die Objekte für einen Zeitraum von 30 Tagen. Danach erfolgt nur noch ein seltener Zugriff für bis zu 90 Tage. Anschließend werden die Objekte nicht mehr benötigt, daher können Sie entscheiden, sie zu archivieren oder zu löschen.

In diesem Szenario erstellen Sie eine S3-Lebenszyklusregel, in der Sie eine erste Überführungsaktion zu S3 Intelligent-Tiering, S3 Standard-IA oder S3 One Zone-IA, eine weitere Überführungsaktion zu S3 Glacier Flexible Retrieval zu Archivierungszwecken und schließlich eine Ablaufaktion angeben. Wenn Sie die Objekte von einer Speicherklasse in eine andere verschieben, sparen Sie Speicherkosten. Weitere Information zu Kostenaspekten finden Sie unter [Amazon S3 – Preise](#).

Themen

- [Übergang von Objekten mit Amazon-S3-Lebenszyklus](#)
- [Auslaufende Objekte](#)
- [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#)
- [Lebenszyklus- und andere Bucket-Konfigurationen](#)
- [Konfigurieren von Lebenszyklus-Ereignisbenachrichtigungen](#)
- [Elemente der Lebenszyklus-Konfiguration](#)
- [Beispiele der S3-Lebenszyklus-Konfiguration](#)

Übergang von Objekten mit Amazon-S3-Lebenszyklus

Sie können Regeln zu Ihrer S3-Lebenszykluskonfiguration hinzufügen und damit Amazon S3 anweisen, Objekte in eine andere Amazon-S3-Speicherklasse zu übergeben. Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#). Einige Beispiele dafür, wann Sie S3-Lebenszykluskonfigurationen auf diese Weise verwenden könnten, sind die folgenden:

- Wenn Sie wissen, dass nur selten auf bestimmte Objekte zugegriffen wird, können Sie diese in die Speicherklasse S3 Standard-IA übergeben.
- Möglicherweise möchten Sie Objekte, auf die Sie nicht zugreifen müssen, in Echtzeit in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archivieren.

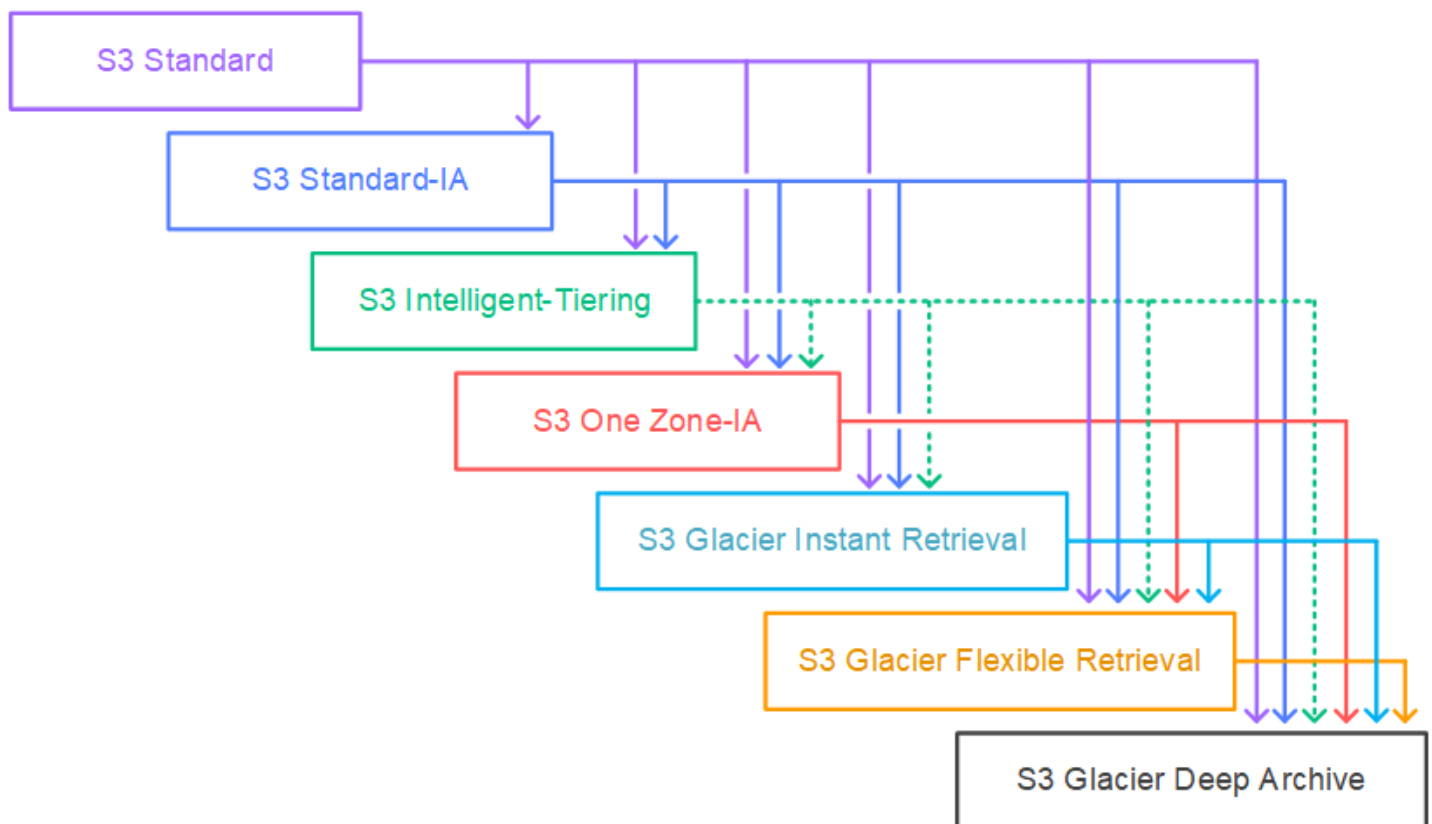
Note

Verschlüsselte Objekte bleiben während des gesamten Übergangsprozesses der Speicherklasse verschlüsselt.

Unterstützte Übergänge

Sie können in einer S3-Lebenszyklus-Konfiguration Regeln definieren, um Objekte aus einer Speicherklasse zu einer anderen Speicherklasse zu überführen, um Speicherkosten zu sparen. Wenn Sie die Zugriffsmuster Ihrer Objekte nicht kennen oder sich Ihre Zugriffsmuster über die Zeit verändern, können Sie die Objekte zur Speicherklasse S3 Intelligent-Tiering überführen, um automatische Kosteneinsparungen zu erzielen. Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Amazon S3 unterstützt das Wasserfallmodell für die Überführung zwischen Speicherklassen wie im folgenden Diagramm gezeigt.



Unterstützte Lebenszyklusübergänge

Amazon S3 unterstützt die folgenden Lebenszyklusübergänge zwischen Speicherklassen mittels einer S3-Lebenszykluskonfiguration.

Die folgenden Übergänge sind möglich:

- Von der Speicherklasse S3 Standard zu einer beliebigen anderen Speicherklasse.
- Von der Speicherklasse S3-Standard-IA zu den Speicherklassen S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.
- Von der Speicherklasse S3 Intelligent-Tiering zu den Speicherklassen S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.

Note

Es gibt einige Ausnahmen für die Übertragung von Objekten von der Speicherklasse S3 Intelligent-Tiering auf S3 One Zone-IA und einige S3 Glacier-Speicherklassen.

Weitere Informationen finden Sie unter [the section called “Nicht unterstützte Lebenszyklusübergänge”](#).

- Von der Speicherklasse S3 One Zone-IA zu den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.
- Von der Speicherklasse S3 Glacier Instant Retrieval zu den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.
- Die Speicherklasse S3 Glacier Flexible Retrieval zur Speicherklasse S3 Glacier Deep Archive.
- Von einer beliebigen Speicherklasse zur Speicherklasse S3 Glacier Deep Archive.

Note

Für Lebenszyklusübergänge fallen keine Gebühren für den Datenabruf an. Es fallen jedoch Gebühren pro Anfrage an, wenn Sie Lebenszyklusregeln verwenden PUTCOPY, um Daten in eine beliebige S3-Speicherklasse zu verschieben. Berücksichtigen Sie die Kosten für die Aufnahme oder Übertragung, bevor Sie Objekte in eine beliebige Speicherklasse verschieben. Weitere Information zu Kostenaspekten finden Sie unter [Amazon S3 – Preise](#).

Nicht unterstützte Lebenszyklusübergänge

Amazon S3 unterstützt keine der folgenden Lebenszyklusübergänge.

Die folgenden Übergänge sind nicht möglich:

- Buckets mit aktivierter Versionierung oder angehaltener Versionierung: alle Objekte mit einem Replikationsstatus. Pending
- Von jeder Speicherklasse zur Speicherklasse S3 Standard.
- Beliebige Speicherklasse der Klasse Reduced Redundancy Storage (). RRS
- Von der S3-One-Zone-IA-Speicherklasse zu den Speicherklassen S3 Intelligent-Tiering, S3 Standard-IA und S3 Glacier Instant Retrieval.
- Von der Speicherklasse S3 Intelligent-Tiering (alle Stufen) zur Speicherklasse S3 Standard-IA.
- Die Archive-Instant-Access-Speicherklasse S3 Intelligent-Tiering-Speicherklasse entspricht der Stufe S3 One Zone-IA.
- Die Archive-Access-Stufe der Speicherklasse S3 Intelligent-Tiering für S3 One Zone-IA oder S3 Glacier Instant Retrieval.
- Die Deep Archive Access-Stufe der Speicherklasse S3 Intelligent-Tiering umfasst S3 One Zone-IA, S3 Glacier Instant Retrieval oder S3 Glacier Flexible Retrieval.

Einschränkungen und Überlegungen für Übergänge

Für die Übergänge der Lebenszyklus-Speicherklassen gelten folgende Einschränkungen:

Objekte, die kleiner als 128 KB sind, werden standardmäßig nicht in eine Speicherklasse umgewandelt

Amazon S3 wendet auf S3-Lifecycle-Konfigurationen ein Standardverhalten an, das verhindert, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden. Wir empfehlen nicht, Objekte mit weniger als 128 KB zu übertragen, da Ihnen für jedes Objekt eine Übergangsanfrage in Rechnung gestellt wird. Das bedeutet, dass bei kleineren Objekten die Umstellungskosten die Speichereinsparungen überwiegen können. Weitere Informationen zu den Kosten für Umstellungsanfragen finden Sie unter Anfragen und Datenabrufe auf der [Amazon S3 S3-Preissseite](#) auf der Registerkarte Speicher und Anfragen.

Um kleineren Objekten den Übergang zu ermöglichen, können Sie Ihren Lebenszyklus-Übergangsregeln einen [Objektgrößenfilter](#) hinzufügen, der eine benutzerdefinierte Mindestgröße

(`ObjectSizeGreaterThan`) oder Maximalgröße (`ObjectSizeLessThan`) festlegt. Weitere Informationen finden Sie unter [Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind](#).

Note

Im September 2024 hat Amazon S3 das standardmäßige Übergangsverhalten für kleine Objekte wie folgt aktualisiert:

- Aktualisiertes Standard-Übergangsverhalten — Ab September 2024 verhindert das Standardverhalten, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden.
- Bisheriges Standard-Übergangsverhalten — Vor September 2024 erlaubte das Standardverhalten, dass Objekte, die kleiner als 128 KB waren, nur in die Speicherklassen S3 Glacier und S3 Glacier Deep Archive übertragen wurden.

Konfigurationen, die vor September 2024 erstellt wurden, behalten das vorherige Übergangsverhalten bei, sofern Sie sie nicht ändern. Das heißt, wenn Sie Regeln erstellen, bearbeiten oder löschen, ändert sich das standardmäßige Übergangsverhalten für Ihre Konfiguration in das aktualisierte Verhalten. Falls es Ihr Anwendungsfall erfordert, können Sie das standardmäßige Übergangsverhalten so ändern, dass Objekte, die kleiner als 128 KB sind, auf S3 Glacier und S3 Glacier Deep Archive umgestellt werden. Verwenden Sie dazu den optionalen `x-amz-transition-object-size-minimum-default` Header in einer [PutBucketLifecycleConfiguration](#)Anfrage.

Objekte müssen mindestens 30 Tage lang gespeichert werden, bevor sie auf S3 Standard-IA oder S3 One Zone-IA umgestellt werden

Bevor Sie Objekte in S3 Standard-IA oder S3 One Zone-IA überführen, müssen diese mindestens 30 Tage in Amazon S3 gespeichert worden sein. Beispielsweise können Sie keine Lebenszyklusregel erstellen, mit der Objekte einen Tag nach dem Erstellungsdatum in die Speicherklasse S3 Standard-IA übergehen. Amazon S3 unterstützt diesen Übergang erst nach 30 Tagen, da auf neuere Objekte meist häufiger zugegriffen wird oder sie schneller gelöscht werden, als das für S3 Standard-IA- oder S3 One Zone-IA-Speicher sinnvoll wäre.

Beim Übergang von nicht aktuellen Objekten (in versionsfähigen Buckets) können dementsprechend nur Objekte, die seit mindestens 30 Tagen nicht mehr aktuell sind, in S3 Standard-IA- oder S3 One

Zone-IA-Speicher übergehen. Eine Liste der Mindestspeicherdauer für alle Speicherklassen finden Sie unter [Vergleich der Amazon-S3-Speicherklassen](#)

Ihnen wird die Übertragung von Objekten vor Ablauf ihrer Mindestspeicherdauer in Rechnung gestellt

Bestimmte Speicherklassen haben eine Mindestdauer für die Speicherung von Objekten. Wenn Sie Objekte vor Ablauf der Mindestdauer aus diesen Speicherklassen herausnehmen, wird Ihnen der Rest dieser Dauer in Rechnung gestellt. Weitere Informationen darüber, welche Speicherklassen eine Mindestspeicherdauer haben, finden Sie unter [Vergleich der Amazon-S3-Speicherklassen](#).

Sie können keine einzelne Lebenszyklusregel erstellen, die Objekte vor Ablauf der Mindestspeicherdauer von einer Speicherklasse in eine andere überträgt.

Beispielsweise hat S3 Glacier Instant Retrieval eine Mindestspeicherdauer von 90 Tagen. Sie können keine Lebenszyklusregel angeben, die Objekte nach 4 Tagen in S3 Glacier Instant Retrieval überführt und Objekte dann nach 20 Tagen in S3 Glacier Deep Archive überträgt. In diesem Fall muss die Umstellung auf S3 Glacier Deep Archive nach mindestens 94 Tagen erfolgen.

Sie können dafür zwei Regeln angeben, zahlen jedoch die Speichergebühren für die Mindestdauer. Weitere Information zu Kostenaspekten finden Sie unter [Amazon S3 – Preise](#).

Weitere Informationen zum Erstellen eines S3-Lebenszyklus finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#).

Übergang in die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive (Objektarchivierung)

Mithilfe einer S3 Lifecycle-Konfiguration können Sie Objekte zur Archivierung in die Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive übertragen.

Bevor Sie Ihre Objekte archivieren, lesen Sie die folgenden Abschnitte, wo Sie weitere relevante Aspekte finden.

Allgemeine Überlegungen

Bevor Sie Objekte archivieren, sollten Sie die folgenden allgemeinen Überlegungen in Betracht ziehen:

- Verschlüsselte Objekte bleiben während des gesamten Übergangsprozesses der Speicherklasse verschlüsselt.

- Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert sind, sind nicht in Echtzeit verfügbar.

Archivierte Objekte sind Amazon-S3-Objekte, aber um auf ein archiviertes Objekt zugreifen zu können, müssen Sie zuerst eine temporäre Kopie davon wiederherstellen. Die wiederhergestellte Objektkopie ist nur für die Dauer verfügbar, die Sie in der Wiederherstellungsanforderung angeben. Danach löscht Amazon S3 die temporäre Kopie und das Objekt bleibt in S3 Glacier Flexible Retrieval archiviert.

Sie können ein Objekt mithilfe der Amazon S3 S3-Konsole oder programmgesteuert mithilfe der AWS SDK Wrapper-Bibliotheken oder Amazon S3 REST API in Ihrem Code. Weitere Informationen finden Sie unter [Wiederherstellen eines archivierten Objekts](#).

- Objekte, die in der Speicherklasse S3 Glacier Flexible Retrieval gespeichert sind, können nur in die Speicherklasse S3 Glacier Deep Archive überführt werden.

Sie können mit einer S3-Lebenszyklusregel die Speicherklasse eines Objekts von S3 Glacier Flexible Retrieval nur zur Speicherklasse S3 Glacier Deep Archive ändern. Wenn Sie die Speicherklasse eines in S3 Glacier Flexible Retrieval gespeicherten Objekts zu einer anderen Speicherklasse als S3 Glacier Deep Archive ändern möchten, müssen Sie die Wiederherstellungsoperation verwenden, um zunächst eine temporäre Kopie des Objekts zu erstellen. Anschließend verwenden Sie die Kopieroperation, um das Objekt zu überschreiben und dabei S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA oder Reduced Redundancy als Speicherklasse anzugeben.

- Die Überführung von Objekten in die Speicherklasse S3 Glacier Deep Archive ist nicht umkehrbar.

Sie können nicht mit einer S3-Lebenszyklusregel die Speicherklasse eines Objekts von S3 Glacier Deep Archive zu einer anderen Speicherklasse ändern. Wenn Sie die Speicherklasse eines archivierten Objekts in eine andere Speicherklasse ändern möchten, müssen Sie die Wiederherstellungsoperation verwenden, um zunächst eine temporäre Kopie des Objekts zu erstellen. Verwenden Sie dann den Kopiervorgang, um das Objekt zu überschreiben, indem Sie S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder Reduced Redundancy Storage als Speicherklasse angeben.

Note

Die Copy-Operation für wiederhergestellte Objekte wird in der Amazon-S3-Konsole für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive

nicht unterstützt. Verwenden Sie für diese Art von Kopiervorgang den AWS Command Line Interface (AWS CLI), das AWS SDKs, oder der RESTAPI.

Die Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive gespeichert sind, sind nur über Amazon S3 sichtbar und verfügbar. Sie sind nicht über den separaten Amazon-S3-Glacier-Service verfügbar.

Dies sind Amazon S3-Objekte, auf die Sie nur über die Amazon S3 S3-Konsole oder Amazon S3 zugreifen könnenAPI. Sie können nicht über die separate Amazon S3 Glacier-Konsole oder Amazon S3 Glacier auf die archivierten Objekte zugreifenAPI.

Kostenüberlegungen

Wenn Sie vorhaben, Daten mit seltenem Zugriff für einen Zeitraum von Monaten oder Jahren zu archivieren, können die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive Ihre Speicherkosten reduzieren. Sie sollten jedoch Folgendes berücksichtigen, um sicherzustellen, dass die Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive geeignet für Sie ist:

- Gebühren für zusätzlichen Speicheraufwand – Wenn Sie Objekte zur Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive überführen, wird jedem Objekt eine feste Speichermenge hinzugefügt, um die Metadaten für die Verwaltung des Objekts zu berücksichtigen.
- Für jedes in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archivierte Objekt verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Amazon S3 speichert diese Metadaten, sodass Sie mithilfe von Amazon S3 eine Echtzeitliste Ihrer archivierten Objekte abrufen könnenAPI. Weitere Informationen finden Sie unter [Get Bucket \(List Objects\)](#). Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt.
- Für jedes Objekt, das in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert wird, fügt Amazon S3 32 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die Gebühren für S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive in Rechnung gestellt.

Wenn Sie kleine Objekte archivieren, sollten Sie diese Speichergebühren berücksichtigen. Ziehen Sie auch in Betracht, viele kleine Objekte in wenigen großen Objekten zusammenzufassen, um Kosten für den Verwaltungsaufwand zu reduzieren.

- Anzahl der Tage, für die die Objekte archiviert werden sollen – S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind langfristige Archivierungslösungen. Die minimale Speicherdauer beträgt 90 Tage für die Speicherklasse S3 Glacier Flexible Retrieval und 180 Tage für S3 Glacier Deep Archive. Beim Löschen von Daten, die in Amazon S3 Glacier archiviert wurden, fallen keine Gebühren an, wenn die von Ihnen gelöschten Objekte länger als die minimale Speicherdauer archiviert werden. Wenn Sie ein archiviertes Objekt innerhalb der minimalen Speicherdauer löschen oder überschreiben, stellt Amazon S3 eine anteilige Gebühr für das vorzeitige Löschen in Rechnung. Weitere Informationen zu den Kosten für die vorzeitige Löschungen finden Sie unter der Frage „Wie wird das Löschen von Objekten aus Amazon S3 Glacier berechnet, die weniger als 90 Tage alt sind?“ Frage zum [Amazon S3 FAQ](#).
- Gebühren für Übergangsanfragen von S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive – Jedes Objekt, das Sie auf die Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive umstellen, stellt eine Übergangsanfrage dar. Für jede dieser Anforderungen entstehen Kosten. Wenn Sie vorhaben, sehr viele Objekte zu überführen, sollten Sie die Anforderungskosten in Betracht ziehen. Wenn Sie eine Mischung von Objekten archivieren, die kleine Objekte enthält, insbesondere solche unter 128 KB, empfehlen wir, den Lifecycle-Objektgrößenfilter zu verwenden, um kleine Objekte aus Ihrer Umstellung herauszufiltern, um die Anforderungskosten zu senken. S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive blockieren nicht automatisch den Übergang von Objekten unter 128 KB.
- Datenwiederherstellungsgebühren für S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive – S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive wurden für die Langzeitarchivierung von Daten entwickelt, auf die Sie selten zugreifen. Informationen zu Gebühren für die Datenwiederherstellung finden Sie unter der Frage „Wie viel kostet das Abrufen von Daten aus Amazon S3 Glacier?“ Frage zum [Amazon S3 FAQ](#). Informationen zum Wiederherstellen von Daten aus Amazon S3 Glacier finden Sie unter [Wiederherstellen eines archivierten Objekts](#).

Wenn Sie Objekte mithilfe der S3-Lebenszyklusverwaltung in Amazon S3 Glacier archivieren, führt Amazon S3 diese Objekte asynchron über. Es kann eine Verzögerung zwischen dem Übergangsdatum in der S3-Lebenszyklus-Konfigurationsregel und dem Datum des physischen Übergangs geben. Ihnen werden die Preise von Amazon S3 Glacier basierend auf dem in der Regel angegebenen Übergangsdatum berechnet. Weitere Informationen finden Sie im Abschnitt Amazon S3 Glacier von [Amazon S3 FAQ](#).

Die Amazon-S3-Produktdetailseite enthält Preisinformationen und Beispielberechnungen für die Archivierung von Amazon-S3-Objekten. Weitere Informationen finden Sie unter den folgenden Themen:

- Wie werden die Speicherkosten für in Amazon S3 Glacier archivierte Amazon-S3-Objekte berechnet? auf dem [Amazon S3 FAQ](#).
- „Wie wird mir das Löschen von Objekten aus Amazon S3 Glacier in Rechnung gestellt, die weniger als 90 Tage alt sind?“ auf dem [Amazon S3 FAQ](#).
- „Was kostet das Abrufen von Daten aus Amazon S3 Glacier?“ auf dem [Amazon S3 FAQ](#).
- [Amazon S3 – Preise](#) für die Speicherkosten der unterschiedlichen Speicherklassen.

Wiederherstellen archivierter Objekte

Auf archivierte Objekte kann nicht in Echtzeit zugegriffen werden. Sie müssen zuerst eine Anforderung zur Wiederherstellung initiieren und dann warten, bis eine temporäre Kopie des Objekts für die Dauer bereitgestellt wird, die Sie in der Anforderung angegeben haben. Nachdem Sie eine temporäre Kopie des wiederhergestellten Objekts erhalten haben, bleibt die Speicherkategorie des Objekts S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive. (A) [HeadObject](#) oder [GetObjectAPI](#)Die Betriebsanforderung gibt S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive als Speicherkategorie zurück.)

Note

Bei einer Wiederherstellung eines Archivs zahlen Sie sowohl für das Archiv (Tarif für S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive) als auch für die temporär wiederhergestellte Kopie (S3-Standard-Speichertarif). Informationen zu Preisen finden Sie unter [Amazon S3 – Preise](#).

Für den Zugriff auf ein kopiertes Objekt müssen Sie eine Anfrage zur Wiederherstellung initiieren. Das kann programmgesteuert oder über die Amazon-S3-Konsole erfolgen. Amazon S3 verarbeitet jeweils nur eine Anfrage pro Objekt. Weitere Informationen finden Sie unter [Wiederherstellen eines archivierten Objekts](#).

Auslaufende Objekte

Wenn ein Objekt aufgrund seiner Lebenszykluskonfiguration das Ende seiner Lebensdauer erreicht, ergreift Amazon S3 eine Aktion, die darauf basiert, in welchem [S3-Versionierungsstatus](#) sich der Bucket befindet.

- Bucket ohne Versionsnummer — Amazon S3 stellt das Objekt zur Entfernung in die Warteschlange und entfernt es asynchron, wodurch das Objekt dauerhaft entfernt wird.
- Bucket mit aktiviertem Versioning – Wenn die aktuelle Objektversion keine Löschmarkierung ist, fügt Amazon S3 die Löschmarkierung mit einer eindeutigen Versions-ID hinzu. Damit ist die aktuelle Version nicht mehr aktuell und die Löschmarkierung wird zur aktuellen Version.
- Bucket mit ausgesetztem Versioning – Amazon S3 erstellt eine Löschmarkierung mit der Versions-ID null. Diese Löschmarkierung ersetzt jede Objektversion mit einer Versions-ID von null in der Versionshierarchie, womit das Objekt effektiv gelöscht wird.

Für einen versionsfähigen Bucket (d. h., das Versioning ist aktiviert oder ausgesetzt) gibt es mehrere Aspekte, die bestimmen, wie Amazon S3 die Ablaufaktion verarbeitet. Für Buckets mit aktivierter oder ausgesetzter Versionsverwaltung gilt Folgendes:

- Die Objektablaufaktion wird nur auf die aktuelle Version eines Objekts angewendet (sie wirkt sich nicht auf nicht aktuelle Objektversionen aus).
- Amazon S3 führt keine Aktion aus, wenn es eine oder mehrere Objektversionen gibt und die Löschmarkierung die aktuelle Version ist.
- Wenn die aktuelle Objektversion die einzige Objektversion und auch eine Löschmarkierung ist (auch als Löschmarkierung eines abgelaufenen Objekts bezeichnet, wobei alle Objektversionen gelöscht werden und nur noch eine Löschmarkierung beibehalten wird), entfernt Amazon S3 die Löschmarkierung des abgelaufenen Objekts. Sie können die Ablaufaktion auch verwenden, um Amazon S3 anzuweisen, alle abgelaufenen Löschmarkierungen zu entfernen. Ein Beispiel finden Sie unter [Beispiel 7: Löschen abgelaufener Löschmarkierungen für Objekte](#).
- Sie können das `NoncurrentVersionExpiration` Action-Element verwenden, um Amazon S3 anzuweisen, nicht aktuelle Versionen von Objekten dauerhaft zu löschen. Diese gelöschten Objekte können nicht wiederhergestellt werden. Sie können dieses Ablaufdatum auf eine bestimmte Anzahl von Tagen beziehen, seit die Objekte nicht mehr aktuell sind. Zusätzlich zur Anzahl der Tage können Sie auch eine maximale Anzahl an veralteten Versionen angeben, die aufbewahrt werden sollen (zwischen 1 und 100). Dieser Wert gibt die Anzahl der neueren nicht aktuellen Versionen an, die vorhanden sein müssen, damit Amazon S3 die zugehörige

Aktion für eine bestimmte Version ausführen kann. Um die maximale Anzahl an nicht aktuellen Versionen anzugeben, müssen Sie auch ein Filter Element angeben. Wenn Sie kein Filter Element angeben, generiert Amazon S3 einen InvalidRequest Fehler, wenn Sie eine maximale Anzahl von nicht aktuellen Versionen angeben. Weitere Informationen zur Verwendung des NoncurrentVersionExpiration Action-Elements finden Sie unter [the section called “Elemente, die Lebenszyklus-Aktionen beschreiben”](#).

- Amazon S3 ergreift keine Maßnahmen für nicht aktuelle Versionen von Objekten, auf die die S3 Object Lock-Konfiguration angewendet wurde.
- Bei Objekten mit einem Pending Replikationsstatus ergreift Amazon S3 keine Maßnahmen für aktuelle oder nicht aktuelle Versionen von Objekten.

Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Important

Wenn Sie mehrere Regeln in einer S3-Lifecycle-Konfiguration haben, kann ein Objekt für mehrere S3 Lifecycle-Aktionen am selben Tag in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von [Löschmarkierungen](#).
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder eine S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht](#).

Bestehende und neue Objekte

Wenn Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie beispielsweise heute eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte mit einem bestimmten Präfix 30 Tage nach ihrer Erstellung ablaufen, stellt Amazon S3 alle

vorhandenen Objekte, die älter als 30 Tage sind und das angegebene Präfix haben, zur Entfernung in eine Warteschlange.

Important

Sie können keine Bucket-Richtlinie verwenden, um Löschungen oder Übergänge durch eine S3-Lebenszyklusregel zu verhindern. Selbst wenn Ihre Bucket-Richtlinie beispielsweise alle Aktionen für alle Principals ablehnt, funktioniert Ihre S3-Lifecycle-Konfiguration weiterhin wie gewohnt.

So finden Sie heraus, wann Objekte ablaufen

Um herauszufinden, wann die aktuelle Version eines Objekts ablaufen soll, verwenden Sie den [HeadObject](#) oder [GetObject](#) API Operation. Diese API Operationen geben Antwortheader zurück, die das Datum und die Uhrzeit angeben, zu dem die aktuelle Version des Objekts nicht mehr zwischengespeichert werden kann.

Note

- Es kann eine Verzögerung zwischen dem Ablaufdatum und dem Datum geben, an dem Amazon S3 ein Objekt entfernt. Der Ablauf oder die mit einem abgelaufenen Objekt verbundene Speicherdauer wird Ihnen nicht in Rechnung gestellt.
- Verwenden Sie vor dem Aktualisieren, Deaktivieren oder Löschen von Lebenszyklusregeln die folgenden LIST API Operationen (wie [ListObjectsV2](#), [ListObjectVersions](#), und [ListMultipartUploads](#)) oder [Amazon S3 Inventory](#) um zu überprüfen, ob Amazon S3 aufgrund Ihrer Anwendungsfälle umgestellt wurde und berechnete Objekte abgelaufen sind.

Gebühren für Mindestspeicherdauer

Wenn Sie eine S3-Lebenszyklusablaufregel erstellen, mit der Objekte ablaufen, die sich seit weniger als 30 Tagen in den Speicherklassen S3 Standard-IA oder S3 One Zone-IA befinden, werden Ihnen Gebühren für 30 Tage in Rechnung gestellt. Wenn Sie eine Lebenszyklusablaufregel erstellen, mit der Objekte ablaufen, die weniger als 90 Tage im Speicher S3 Glacier Flexible Retrieval gewesen sind, werden Ihnen Gebühren für 90 Tage in Rechnung gestellt. Wenn Sie eine Lebenszyklusablaufregel erstellen, mit der Objekte ablaufen, die weniger als 180 Tage im S3 Glacier Deep Archive-Speicher gewesen sind, werden Ihnen Gebühren für 180 Tage in Rechnung gestellt.

Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Einstellung einer Lebenszykluskonfiguration für einen Bucket

In diesem Abschnitt wird erklärt, wie Sie mithilfe der Amazon S3-Konsole eine Amazon S3-Lifecycle-Konfiguration für einen Bucket einrichten können, AWS Command Line Interface (AWS CLI), das AWS SDKs, oder der Amazon S3 RESTAPI. Weitere Informationen zur S3-Lebenszyklus-Konfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Mit Lebenszyklusregeln können Sie Aktionen definieren, die Amazon S3 während der Lebensdauer eines Objekts ausführen soll (z. B. die Überführung von Objekten in eine andere Speicherklasse, ihre Archivierung oder ihr Löschen nach einem bestimmten Zeitraum).

Bevor Sie eine Lebenszyklus-Konfiguration festlegen, beachten Sie Folgendes:

Verzögerung bei der Übertragung der Lebenszykluskonfiguration

Wenn Sie eine S3-Lebenszyklus-Konfiguration zu einem Bucket hinzufügen, tritt eine gewisse Verzögerung ein, bis eine neue oder aktualisierte Lebenszyklus-Konfiguration vollständig auf alle Amazon-S3-Systeme verteilt ist. Rechnen Sie mit einer Verzögerung von einigen Minuten, bis die Konfiguration vollständig wirksam ist. Diese Verzögerung kann auch auftreten, wenn Sie eine S3-Lebenszyklus-Konfiguration löschen.

Verzögerung bei Übergang oder Ablauf

Es gibt eine Verzögerung zwischen dem Zeitpunkt, an dem eine Lebenszyklusregel erfüllt ist, und dem Abschluss der Aktion für die Regel. Nehmen wir beispielsweise an, dass eine Gruppe von Objekten aufgrund einer Lebenszyklusregel am 1. Januar abgelaufen ist. Obwohl die Ablaufregel am 1. Januar erfüllt wurde, löscht Amazon S3 diese Objekte möglicherweise erst Tage oder sogar Wochen später. Diese Verzögerung tritt auf, weil S3 Lifecycle Objekte für Übergänge oder Ablaufzeiten asynchron in die Warteschlange stellt. Änderungen an der Abrechnung werden jedoch normalerweise angewendet, wenn die Lebenszyklusregel erfüllt ist, auch wenn die Aktion noch nicht abgeschlossen ist. Weitere Informationen finden Sie unter [Änderungen bei der Abrechnung](#). Informationen zur Überwachung der Auswirkungen von Aktualisierungen, die durch aktive Lebenszyklusregeln vorgenommen wurden, finden Sie unter [the section called “Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?”](#)

Lebenszyklusregeln aktualisieren, deaktivieren oder löschen

Wenn Sie Lebenszyklusregeln deaktivieren oder löschen, beendet Amazon S3 nach einer kleinen Verzögerung die Planung neuer Objekte für das Löschen oder den Übergang. Die Planung aller bereits geplanten Objekte wird aufgehoben und sie werden nicht gelöscht oder überführt.

Note

Bevor Sie Lebenszyklusregeln aktualisieren, deaktivieren oder löschen, verwenden Sie die folgenden LIST API Operationen (wie [ListObjectsV2](#), [ListObjectVersions](#), und [ListMultipartUploads](#)) oder [Amazon S3 Inventory](#) um zu überprüfen, ob Amazon S3 aufgrund Ihrer Anwendungsfälle umgestellt wurde und berechnete Objekte abgelaufen sind. Falls Sie Probleme beim Aktualisieren, Deaktivieren oder Löschen von Lebenszyklusregeln haben, finden Sie weitere Informationen unter [Behebung von Problemen mit Amazon S3 Lifecycle](#)

Bestehende und neue Objekte

Wenn Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie beispielsweise heute eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte mit einem bestimmten Präfix 30 Tage nach ihrer Erstellung ablaufen, stellt Amazon S3 alle vorhandenen Objekte, die älter als 30 Tage sind und das angegebene Präfix haben, zur Entfernung in eine Warteschlange.

Überwachung der Auswirkungen von Lebenszyklusregeln

Informationen zur Überwachung der Auswirkungen von Aktualisierungen, die durch aktive Lebenszyklusregeln vorgenommen wurden, finden Sie unter [the section called “Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?”](#)

Änderungen bei der Abrechnung

Möglicherweise besteht eine Verzögerung zwischen dem Zeitpunkt, an dem die Lebenszyklus-Konfigurationsregeln erfüllt sind, und dem Zeitpunkt, zu dem die Aktion ausgeführt wird, die durch die Erfüllung der Regel ausgelöst wurde. Änderungen an der Abrechnung werden jedoch vorgenommen, sobald die Lifecycle-Konfigurationsregel erfüllt ist, auch wenn die Maßnahme noch nicht ergriffen wurde.

Beispielsweise wird Ihnen nach Ablauf des Objektlaufs kein Speicherplatz in Rechnung gestellt, auch wenn das Objekt nicht sofort gelöscht wird. Ebenso werden Ihnen nach Ablauf der

Objektübergangszeit die Speichertarife für S3 Glacier Flexible Retrieval berechnet, auch wenn das Objekt nicht sofort in die Speicherklasse S3 Glacier Flexible Retrieval umgestellt wird.

Lebenszyklusübergänge zur Speicherklasse S3 Intelligent-Tiering sind jedoch die Ausnahme. Änderungen an der Abrechnung werden erst vorgenommen, nachdem das Objekt in die S3 Intelligent-Tiering-Speicherklasse übergegangen ist.

Mehrere oder widersprüchliche Regeln

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von [Löschmarkierungen](#).
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder eine S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht](#).

Verwenden der S3-Konsole

Sie können Lebenszyklusregeln für alle Objekte oder eine Teilmenge von Objekten in einem Bucket definieren, indem Sie ein gemeinsames Präfix (Objektnamen, die mit einer gemeinsamen Zeichenfolge beginnen) oder ein Tag verwenden. In Ihrer Lebenszyklusregel können Sie Aktionen definieren, die für aktuelle und nicht aktuelle Objektversionen spezifisch sind. Weitere Informationen finden Sie hier:

- [Verwaltung des Lebenszyklus von Objekten](#)
- [Verwenden der Versioning in S3-Buckets](#)

So erstellen Sie eine Lebenszyklusregel:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.


2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Lebenszyklusregel erstellen möchten.
3. Wählen Sie den Tab Management (Verwaltung) und dann die Option Create lifecycle rule (Lebenszyklusregel erstellen).
4. Geben Sie unter Lifecycle rule name (Name der Lebenszyklusregel) einen Namen für Ihre Regel ein.

Der Name muss innerhalb des Buckets eindeutig sein.

5. Wählen Sie den Umfang der Lebenszyklusregel:
 - Um diese Lebenszyklusregel auf alle Objekte mit einem bestimmten Präfix oder Tag anzuwenden, wählen Sie Umfang auf bestimmte Präfixe oder Markierungen beschränken aus.
 - Um den Bereich nach Präfix zu beschränken, geben Sie unter Prefix (Präfix) das Präfix ein.
 - Um den Bereich nach Tag einzuschränken, wählen Sie Add tag (Tag hinzufügen), und geben Sie den Tag-Schlüssel und den Wert ein.

Weitere Hinweise zu Präfixen für Objektnamen finden Sie unter [Erstellen von Objektschlüsselnamen](#). Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

- Um diese Lebenszyklusregel auf alle Objekte im Bucket anzuwenden, wählen Sie Diese Regel gilt für alle Objekte im Bucket und anschließend Ich bestätige, dass diese Regel für alle Objekte im Bucket gilt.
6. Um eine Regel nach Objektgröße zu filtern, können Sie „Minimale Objektgröße angeben“, „Maximale Objektgröße angeben“ oder beide Optionen auswählen.
 - Wenn Sie einen Wert für Minimale Objektgröße oder Maximale Objektgröße angeben, muss der Wert größer als 0 Byte und bis zu 5 TB sein. Sie können diesen Wert in Byte, KB, MB oder GB angeben.
 - Wenn Sie beide Werte angeben, muss die maximale Objektgröße größer als die minimale Objektgröße sein.


 Note

Die Filter Minimale Objektgröße und Maximale Objektgröße schließen die angegebenen Werte aus. Wenn Sie beispielsweise einen Filter so einrichten, dass Objekte mit einer Mindestobjektgröße von 128 KB ablaufen, laufen Objekte, die genau

128 KB groß sind, nicht ab. Stattdessen gilt die Regel nur für Objekte mit einer Größe von mehr als 128 KB.

7. Wählen Sie unter Lifecycle rule actions (Lebenszyklusregelaktionen) die Aktionen aus, die Ihre Lebenszyklusregel ausführen soll:

- Umsetzung aktueller Versionen von Objekten zwischen Speicherklassen
- Umsetzung früherer Versionen von Objekten zwischen Speicherklassen
- Ablauf aktueller Versionen von Objekten

 Note

Bei Buckets, für die die [S3-Versionierung](#) nicht aktiviert ist, führt das Ablaufen aktueller Versionen dazu, dass Amazon S3 die Objekte dauerhaft löscht. Weitere Informationen finden Sie unter [the section called “Lebenszyklus-Aktionen und der Versioning-Status eines Buckets”](#).

- Dauerhaftes Löschen früherer Versionen von Objekten
- Löschen abgelaufener Löschmarkierungen oder unvollständiger mehrteiliger Uploads

Abhängig von den von Ihnen ausgewählten Aktionen werden verschiedene Optionen angezeigt.


8. So setzen Sie aktuelle Versionen von Objekten zwischen Speicherklassen unter Transition current versions of objects between storage classes (Umsetzen von aktuellen Versionen von Objekten zwischen Speicherklassen) um:

a. Wählen Sie unter Speicherklassenübergänge die Speicherklasse aus, zu der Sie wechseln möchten. Eine Liste möglicher Übergänge finden Sie unter [the section called “Unterstützte Lebenszyklusübergänge”](#). Sie können aus den folgenden Speicherklassen wählen:

- S3 Standard-IA
- S3 Intelligent-Tiering
- S3 One Zone-IA
- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive

- b. Geben Sie im Feld Days after object creation (Tage nach der Objekterstellung) die Anzahl der Tage nach der Erstellung für die Umsetzung des Objekts ein.

Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#). Sie können Übergänge für aktuelle oder vorhergehende Objektversionen definieren oder sowohl für aktuelle als auch vorhergehende Versionen. Versioning ermöglicht Ihnen, mehrere Versionen eines Objekts in einem Bucket aufzubewahren. Weitere Informationen über die Versionssteuerung finden Sie unter [Verwenden der S3-Konsole](#).

 **Important**

Wenn Sie die Speicherklasse S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder Glacier Deep Archive wählen, verbleiben Ihre Objekte in Amazon S3. Sie können nicht direkt über den separaten Amazon-S3-Glacier-Service auf sie zugreifen. Weitere Informationen finden Sie unter [Übergang von Objekten mit Amazon-S3-Lebenszyklus](#).

9. Um ältere Versionen von Objekten zwischen Speicherklassen zu übertragen, gehen Sie unter Übergang von nicht aktuellen Versionen von Objekten zwischen Speicherklassen wie folgt vor:
 - a. Wählen Sie unter Speicherklassenübergänge die Speicherklasse aus, zu der der Übergang erfolgen soll. Eine Liste möglicher Übergänge finden Sie unter [the section called "Unterstützte Lebenszyklusübergänge"](#). Sie können aus den folgenden Speicherklassen wählen:
 - S3 Standard-IA
 - S3 Intelligent-Tiering
 - S3 One Zone-IA
 - S3 Glacier Instant Retrieval
 - S3 Glacier Flexible Retrieval
 - S3 Glacier Deep Archive
 - b. Geben Sie im Feld Tage, nachdem das Objekt nicht mehr aktuell ist, die Anzahl der Tage nach der Erstellung ein, um das Objekt umzustellen.
10. Um den Ablauf aktueller Versionen von Objekten auszulösen, geben Sie unter Expire current versions of objects (Ablauf aktueller Objektversionen) in Number of days after object creation (Anzahl der Tage nach der Objekterstellung) die Anzahl der Tage ein.

⚠ Important

In einem Bucket ohne Versionierung führt die Ablaufaktion dazu, dass Amazon S3 das Objekt dauerhaft entfernt. Weitere Informationen über die Lebenszyklus-Aktionen finden Sie unter [Elemente, die Lebenszyklus-Aktionen beschreiben](#).

11. Um frühere Versionen von Objekten dauerhaft zu löschen, geben Sie unter Permanently delete previous versions of objects (Dauerhaftes Löschen nicht aktueller Versionen von Objekten) in Days after objects become noncurrent (Tage nach dem Erlöschen von Objekten) die Anzahl der Tage ein. Sie können die Anzahl der beizubehaltenden neueren Versionen optional angeben, indem Sie einen Wert unter Number of newer versions to retain (Anzahl der beizubehaltenden neueren Versionen) eingeben.
12. Wählen Sie unter Delete expired delete markers or incomplete multipart uploads (Abgelaufene Löschmarkierungen oder unvollständige mehrteilige Uploads löschen) Delete expired object delete markers (Abgelaufene Objektlöschmarken löschen) und Delete incomplete multipart uploads (Unvollständige mehrteilige Uploads löschen) aus. Geben Sie dann die Anzahl der Tage nach der mehrteiligen Upload-Initiierung ein, die Sie beenden und für die Sie unvollständige mehrteilige Uploads löschen möchten.

Weitere Informationen über mehrteilige Uploads finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

13. Wählen Sie Create rule aus.

Wenn die Regel keine Fehler enthält, aktiviert Amazon S3 sie, und Sie können sie im Tab Management (Verwaltung) unter Lifecycle rules (Lebenszyklusregeln) sehen.

Für Informationen über AWS CloudFormation Vorlagen und Beispiele finden Sie unter [Arbeiten mit AWS CloudFormation Vorlagen](#) und [AWS::S3::Bucket](#) in der AWS CloudFormation Benutzerleitfaden.

Verwendung der AWS CLI

Sie können Folgendes verwenden AWS CLI Befehle zur Verwaltung von S3 Lifecycle-Konfigurationen:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`

- `delete-bucket-lifecycle`

Für Anweisungen zur Einrichtung des AWS CLI, finden Sie unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

Die Amazon S3 Lifecycle-Konfiguration ist eine XML Datei. Aber wenn Sie das verwenden AWS CLI, Sie können das XML Format nicht angeben. Sie müssen stattdessen das JSON Format angeben. Im Folgenden finden Sie Beispiele für XML Lebenszykluskonfigurationen und die entsprechenden JSON Konfigurationen, die Sie in einem angeben können AWS CLI Befehl.

Betrachten Sie das folgende Beispiel einer S3-Lebenszyklus-Konfiguration:

Example Beispiel 1

Example

XML

```
<LifecycleConfiguration>
  <Rule>
    <ID>ExampleRule</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

JSON

```
{
  "Rules": [
    {
```

```
    "Filter": {
      "Prefix": "documents/"
    },
    "Status": "Enabled",
    "Transitions": [
      {
        "Days": 365,
        "StorageClass": "GLACIER"
      }
    ],
    "Expiration": {
      "Days": 3650
    },
    "ID": "ExampleRule"
  }
]
```

Example Beispiel 2

Example

XML

```
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ID>id-1</ID>
    <Expiration>
      <Days>1</Days>
    </Expiration>
    <Filter>
      <And>
        <Prefix>myprefix</Prefix>
        <Tag>
          <Key>mytagkey1</Key>
          <Value>mytagvalue1</Value>
        </Tag>
        <Tag>
          <Key>mytagkey2</Key>
          <Value>mytagvalue2</Value>
        </Tag>
      </And>
    </Filter>
  </Rule>
</LifecycleConfiguration>
```

```
<Status>Enabled</Status>
</Rule>
</LifecycleConfiguration>
```

JSON

```
{
  "Rules": [
    {
      "ID": "id-1",
      "Filter": {
        "And": {
          "Prefix": "myprefix",
          "Tags": [
            {
              "Value": "mytagvalue1",
              "Key": "mytagkey1"
            },
            {
              "Value": "mytagvalue2",
              "Key": "mytagkey2"
            }
          ]
        }
      },
      "Status": "Enabled",
      "Expiration": {
        "Days": 1
      }
    }
  ]
}
```

Sie können den Befehl `put-bucket-lifecycle-configuration` wie folgt testen.

So testen Sie die Konfiguration

1. Speichern Sie die JSON Lifecycle-Konfiguration in einer Datei (z. B. *lifecycle.json*).
2. Führen Sie Folgendes aus AWS CLI Befehl, um die Lifecycle-Konfiguration für Ihren Bucket festzulegen. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
$ aws s3api put-bucket-lifecycle-configuration \
--bucket amzn-s3-demo-bucket \
--lifecycle-configuration file:///lifecycle.json
```

3. Rufen Sie zur Überprüfung die S3-Lifecycle-Konfiguration ab, indem Sie den `get-bucket-lifecycle-configuration` AWS CLI Befehl wie folgt:

```
$ aws s3api get-bucket-lifecycle-configuration \
--bucket amzn-s3-demo-bucket
```

4. Um die S3 Lifecycle-Konfiguration zu löschen, verwenden Sie den `delete-bucket-lifecycle` AWS CLI Befehl wie folgt:

```
aws s3api delete-bucket-lifecycle \
--bucket amzn-s3-demo-bucket
```

Verwendung der AWS SDKs

Java

Sie können das AWS SDK for Java um die S3-Lifecycle-Konfiguration eines Buckets zu verwalten. Weitere Informationen zur Verwaltung einer S3-Lebenszyklus-Konfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Note

Wenn Sie einem Bucket eine S3-Lebenszyklus-Konfiguration hinzufügen, ersetzt Amazon S3 die aktuelle Lebenszyklus-Konfiguration des Buckets, sofern vorhanden. Um eine Lebenszyklus-Konfiguration zu aktualisieren, rufen Sie sie ab, nehmen die gewünschten Änderungen vor und fügen die geänderte Konfiguration dann dem Bucket hinzu.

Das folgende Beispiel zeigt die Verwendung von AWS SDK for Java um die Lifecycle-Konfiguration eines Buckets hinzuzufügen, zu aktualisieren und zu löschen. Das Beispiel erledigt Folgendes:

- Fügt eine Lebenszyklus-Konfiguration zu einem Bucket hinzu.
- Ruft die Lebenszyklus-Konfiguration und Updates durch Hinzufügen einer weiteren Regel ab.

- Fügt die abgeänderte Lebenszyklus-Konfiguration dem Bucket hinzu. Amazon S3 ersetzt die vorhandene Konfiguration.
- Ruft die Konfiguration erneut ab und überprüft, ob sie die richtige Anzahl von Regeln enthält, indem die Anzahl der Regeln gedruckt wird.
- Löscht die Lebenszyklus-Konfiguration und überprüft, ob sie gelöscht wurde, indem versucht wird, sie erneut abzurufen.

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration.Transition;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.Tag;
import com.amazonaws.services.s3.model.lifecycle.LifecycleAndOperator;
import com.amazonaws.services.s3.model.lifecycle.LifecycleFilter;
import com.amazonaws.services.s3.model.lifecycle.LifecyclePrefixPredicate;
import com.amazonaws.services.s3.model.lifecycle.LifecycleTagPredicate;

import java.io.IOException;
import java.util.Arrays;

public class LifecycleConfiguration {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";

        // Create a rule to archive objects with the "glacierobjects/"
prefix to Glacier
        // immediately.
        BucketLifecycleConfiguration.Rule rule1 = new
BucketLifecycleConfiguration.Rule()
            .withId("Archive immediately rule")
```

```
        .withFilter(new LifecycleFilter(new
LifecyclePrefixPredicate("glacierobjects/")))
        .addTransition(new
Transition().withDays(0).withStorageClass(StorageClass.Glacier))
        .withStatus(BucketLifecycleConfiguration.ENABLED);

// Create a rule to transition objects to the Standard-Infrequent
Access storage
// class
// after 30 days, then to Glacier after 365 days. Amazon S3 will
delete the
// objects after 3650 days.
// The rule applies to all objects with the tag "archive" set to
"true".
    BucketLifecycleConfiguration.Rule rule2 = new
BucketLifecycleConfiguration.Rule()
        .withId("Archive and then delete rule")
        .withFilter(new LifecycleFilter(new
LifecycleTagPredicate(new Tag("archive", "true"))))
        .addTransition(new Transition().withDays(30)

.withStorageClass(StorageClass.StandardInfrequentAccess))
        .addTransition(new
Transition().withDays(365).withStorageClass(StorageClass.Glacier))
        .withExpirationInDays(3650)
        .withStatus(BucketLifecycleConfiguration.ENABLED);

// Add the rules to a new BucketLifecycleConfiguration.
BucketLifecycleConfiguration configuration = new
BucketLifecycleConfiguration()
        .withRules(Arrays.asList(rule1, rule2));

try {
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .withCredentials(new
ProfileCredentialsProvider())
        .withRegion(clientRegion)
        .build();

// Save the configuration.
s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

// Retrieve the configuration.
```



```
        configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

        // Add a new rule with both a prefix predicate and a tag
predicate.
        configuration.getRules().add(new
BucketLifecycleConfiguration.Rule().withId("NewRule")
                .withFilter(new LifecycleFilter(new
LifecycleAndOperator(
                                Arrays.asList(new
LifecyclePrefixPredicate("YearlyDocuments/"),
                                new
LifecycleTagPredicate(new Tag(
                "expire_after",
                "ten_years"))))))))
                .withExpirationInDays(3650)

.withStatus(BucketLifecycleConfiguration.ENABLED));

        // Save the configuration.
s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

        // Retrieve the configuration.
configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

        // Verify that the configuration now has three rules.
configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);
        System.out.println("Expected # of rules = 3; found: " +
configuration.getRules().size());

        // Delete the configuration.
s3Client.deleteBucketLifecycleConfiguration(bucketName);

        // Verify that the configuration has been deleted by
attempting to retrieve it.
configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);
        String s = (configuration == null) ? "No configuration
found." : "Configuration found.";
```

```
        System.out.println(s);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3
couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the
client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Sie können das AWS SDK for .NET um die S3 Lifecycle-Konfiguration in einem Bucket zu verwalten. Weitere Informationen zur Verwaltung einer Lebenszyklus-Konfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Note

Wenn Sie eine Lebenszyklus-Konfiguration hinzufügen, ersetzt Amazon S3 die vorhandene Konfiguration für den angegebenen Bucket. Um eine Konfiguration zu aktualisieren, müssen Sie zuerst die Lebenszyklus-Konfiguration abrufen, die Änderungen vornehmen und dann die geänderte Lebenszyklus-Konfiguration dem Bucket hinzufügen.

Das folgende Beispiel zeigt die Verwendung von AWS SDK for .NET um die Lifecycle-Konfiguration eines Buckets hinzuzufügen, zu aktualisieren und zu löschen. Das Codebeispiel führt die folgenden Aufgaben durch:

- Fügt eine Lebenszyklus-Konfiguration zu einem Bucket hinzu.
- Ruft die Lebenszyklus-Konfiguration und Updates durch Hinzufügen einer weiteren Regel ab.
- Fügt die abgeänderte Lebenszyklus-Konfiguration dem Bucket hinzu. Amazon S3 ersetzt die vorhandene Lebenszyklus-Konfiguration.
- Ruft die Konfiguration erneut ab und überprüft sie durch Ausgabe der Anzahl von Regeln in der Konfiguration.

- Löscht die Lebenszyklus-Konfiguration und überprüft den Löschvorgang.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class LifecycleTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddUpdateDeleteLifecycleConfigAsync().Wait();
        }

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List<LifecycleRule>
                    {
                        new LifecycleRule
                        {
                            Id = "Archive immediately rule",
                            Filter = new LifecycleFilter()
                            {
                                LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                                {
```

```
        Prefix = "glacierobjects/"
    }
},
Status = LifecycleRuleStatus.Enabled,
Transitions = new List<LifecycleTransition>
{
    new LifecycleTransition
    {
        Days = 0,
        StorageClass = S3StorageClass.Glacier
    }
},
},
new LifecycleRule
{
    Id = "Archive and then delete rule",
    Filter = new LifecycleFilter()
    {
        LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
        {
            Prefix = "projectdocs/"
        }
    },
    Status = LifecycleRuleStatus.Enabled,
    Transitions = new List<LifecycleTransition>
    {
        new LifecycleTransition
        {
            Days = 30,
            StorageClass =
S3StorageClass.StandardInfrequentAccess
        },
        new LifecycleTransition
        {
            Days = 365,
            StorageClass = S3StorageClass.Glacier
        }
    },
    Expiration = new LifecycleRuleExpiration()
    {
        Days = 3650
    }
}
}
```

```
        }
    };

    // Add the configuration to the bucket.
    await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

    // Retrieve an existing configuration.
    lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

    // Add a new rule.
    lifeCycleConfiguration.Rules.Add(new LifecycleRule
    {
        Id = "NewRule",
        Filter = new LifecycleFilter()
        {
            LifecycleFilterPredicate = new LifecyclePrefixPredicate()
            {
                Prefix = "YearlyDocuments/"
            }
        },
        Expiration = new LifecycleRuleExpiration()
        {
            Days = 3650
        }
    });

    // Add the configuration to the bucket.
    await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

    // Verify that there are now three rules.
    lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
    Console.WriteLine("Expected # of rulest=3; found:{0}",
lifeCycleConfiguration.Rules.Count);

    // Delete the configuration.
    await RemoveLifecycleConfigAsync(client);

    // Retrieve a nonexistent configuration.
    lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
}
catch (AmazonS3Exception e)
```

```
        {
            Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
    }

    static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
    {
        PutLifecycleConfigurationRequest request = new
PutLifecycleConfigurationRequest
        {
            BucketName = bucketName,
            Configuration = configuration
        };
        var response = await client.PutLifecycleConfigurationAsync(request);
    }

    static async Task<LifecycleConfiguration>
RetrieveLifecycleConfigAsync(IAmazonS3 client)
    {
        GetLifecycleConfigurationRequest request = new
GetLifecycleConfigurationRequest
        {
            BucketName = bucketName
        };
        var response = await client.GetLifecycleConfigurationAsync(request);
        var configuration = response.Configuration;
        return configuration;
    }

    static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
    {
        DeleteLifecycleConfigurationRequest request = new
DeleteLifecycleConfigurationRequest
        {
            BucketName = bucketName
        };
    }
};
```

```
        await client.DeleteLifecycleConfigurationAsync(request);
    }
}
}
```

Ruby

Sie können das AWS SDK for Ruby um die S3-Lifecycle-Konfiguration in einem Bucket mithilfe der Klasse [AWS::S3::BucketLifecycleConfiguration](#) zu verwalten. Weitere Informationen zur Verwaltung einer Lebenszyklus-Konfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Mit dem REST API

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird die Konfiguration REST API im Zusammenhang mit der S3 Lifecycle-Konfiguration beschrieben.

- [PutBucketLifecycleConfiguration](#)
- [GetBucketLifecycleConfiguration](#)
- [DeleteBucketLifecycle](#)

Fehlerbehebung bei S3 Lifecycle

Informationen zu häufigen Problemen, die bei der Arbeit mit S3 Lifecycle auftreten können, finden Sie unter [the section called “Behebung von Problemen im Lebenszyklus”](#).

Lebenszyklus- und andere Bucket-Konfigurationen

Neben den S3-Lebenszyklus-Konfigurationen können Sie Ihrem Bucket auch weitere Konfigurationen zuordnen. In diesem Abschnitt wird erläutert, wie sich die S3-Lebenszyklus-Konfiguration auf andere Bucket-Konfigurationen auswirkt.

Lebenszyklus und Versioning

Sie können nicht versionsfähigen Buckets und versionsfähigen Buckets S3-Lebenszyklus-Konfigurationen hinzufügen. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Ein versionsfähiger Bucket behält eine aktuelle Objektversion sowie und null oder mehr langfristige Objektversionen bei. Sie können separate Lebenszyklusrichtlinie für aktuelle und nicht aktuelle Objektversionen definieren.

Weitere Informationen finden Sie unter [Elemente der Lebenszyklus-Konfiguration](#).

Important

Wenn Sie mehrere Regeln in einer S3-Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von [Löschmarkierungen](#).
- Wenn ein Objekt sowohl für einen S3 Glacier Flexible Retrieval als auch für einen S3 Standard-IA (oder S3 One Zone-IA) -Übergang in Frage kommt, entscheidet sich Amazon S3 für den Übergang zu S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht](#).

Lebenszykluskonfiguration für Buckets mit MFA aktiviertem Status


Die Lebenszykluskonfiguration für Buckets mit Multi-Faktor-Authentifizierung (MFA) wird nicht unterstützt.

Lebenszyklus und Protokollieren

Amazon S3 Lifecycle-Aktionen werden nicht erfasst von AWS CloudTrail Protokollierung auf Objektebene. CloudTrailerfasst API Anfragen an externe Amazon S3 S3-Endpunkte, wohingegen S3 Lifecycle-Aktionen über interne Amazon S3 S3-Endpunkte ausgeführt werden. Amazon-S3-Server-Zugriffsprotokolle können in einem S3-Bucket aktiviert werden, um S3-Lebenszyklus-Aktionen wie den Objektübergang zu einer anderen Speicherklasse und einen Objektlauf zu erfassen, was zu dauerhaftem Löschen oder logischen Löschen führt. Weitere Informationen finden Sie unter [the section called "Protokollierungs-Serverzugriff"](#).

Wenn die Protokollierung für Ihren Bucket aktiviert ist, melden Amazon-S3-Server-Zugriffsprotokolle die Ergebnisse der folgenden Vorgänge.

Operationsprotokoll	Beschreibung
S3.EXPIRE.OBJECT	Amazon S3 löscht das Objekt aufgrund der Lifecycle-Ablaufaktion dauerhaft.
S3.CREATE.DELETEMARKER	Amazon S3 löscht logisch die aktuelle Version und fügt eine Löschmarkierung in einem Bucket mit aktivierter Versionierung hinzu.
S3.TRANSITION_SIA.OBJECT	Amazon S3 überträgt das Objekt in die Speicherklasse S3 Standard-IA.
S3.TRANSITION_ZIA.OBJECT	Amazon S3 überträgt das Objekt in die Speicherklasse S3 One Zone-IA.
S3.TRANSITION_INT.OBJECT	Amazon S3 überträgt das Objekt in die Speicherklasse S3 Intelligent-Tiering.
S3.TRANSITION_GIR.OBJECT	Amazon S3 initiiert den Übergang des Objekts in die Speicherklasse S3 Glacier Instant Retrieval.
S3.TRANSITION.OBJECT	Amazon S3 initiiert den Übergang des Objekts zur Speicherklasse S3 Glacier Flexible Retrieval.
S3.TRANSITION_GDA.OBJECT	Amazon S3 initiiert den Übergang des Objekts in die Speicherklasse S3 Glacier Deep Archive.
S3.DELETE.UPLOAD	Amazon S3 bricht einen unvollständigen mehrteiligen Upload ab.

 Note

Amazon-S3-Server-Zugriffsprotokolle werden in der Regel auf Best-Effort-Basis bereitgestellt. Sie sind nicht als vollständige Auflistung aller Amazon-S3-Anforderungen vorgesehen.

Fehlerbehebung bei S3 Lifecycle

Weitere Informationen zur Behebung häufiger Probleme mit S3 Lifecycle finden Sie unter [Behebung von Problemen mit Amazon S3 Lifecycle](#).

Weitere Informationen

- [Elemente der Lebenszyklus-Konfiguration](#)
- [Übergang in die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive \(Objektarchivierung\)](#)
- [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#)

Konfigurieren von Lebenszyklus-Ereignisbenachrichtigungen

Sie können eine Amazon-S3-Ereignisbenachrichtigung einrichten, um eine Benachrichtigung zu erhalten, wenn Amazon S3 ein Objekt löscht oder es nach einer S3-Lebenszyklusregel in eine andere Amazon-S3-Speicherklasse überführt.

Mithilfe der `LifecycleExpiration` Ereignistypen können Sie Benachrichtigungen erhalten, wenn Amazon S3 ein Objekt auf der Grundlage Ihrer S3-Lifecycle-Konfiguration löscht. Der Ereignistyp `s3:LifecycleExpiration:Delete` benachrichtigt Sie, wenn ein Objekt in einem unversionierten Bucket gelöscht wird. Es benachrichtigt Sie auch, wenn eine Objektversion durch eine S3-Lebenszyklus-Konfiguration dauerhaft gelöscht wird. Der `s3:LifecycleExpiration:DeleteMarkerCreated` Ereignistyp benachrichtigt Sie, wenn S3 Lifecycle eine Löschmarkierung erstellt, wenn eine aktuelle Version eines Objekts in einem versionierten Bucket gelöscht wird. Weitere Informationen hierzu finden Sie unter [Objektversion löschen](#).

Mithilfe des `s3:LifecycleTransition` Ereignistyps können Sie eine Benachrichtigung erhalten, wenn ein Objekt durch eine S3-Lifecycle-Konfiguration von einer Amazon S3 S3-Speicherklasse in eine andere übertragen wird.

Amazon S3 kann Ereignisbenachrichtigungen zu einem Amazon Simple Notification Service (AmazonSNS) -Thema, einer Amazon Simple Queue Service (AmazonSQS) -Warteschlange oder einer AWS Lambda Funktion veröffentlichen. Weitere Informationen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Anweisungen zum Konfigurieren von Amazon-S3-Ereignisbenachrichtigungen finden Sie unter [Aktivieren von Ereignisbenachrichtigungen](#).

Das Folgende ist ein Beispiel für eine Nachricht, die Amazon S3 sendet, um ein `s3:LifecycleExpiration:Delete`-Ereignis zu veröffentlichen. Weitere Informationen finden Sie unter [Struktur der Ereignisnachricht](#).

```
{
  "Records":[
    {
      "eventVersion":"2.3",
      "eventSource":"aws:s3",
      "awsRegion":"us-west-2",
      "eventTime":"1970-01-01T00:00:00.000Z",
      "eventName":"LifecycleExpiration:Delete",
      "userIdentity":{
        "principalId":"s3.amazonaws.com"
      },
      "requestParameters":{
        "sourceIPAddress":"s3.amazonaws.com"
      },
      "responseElements":{
        "x-amz-request-id":"C3D13FE58DE4C810",
        "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvAN0jpD"
      },
      "s3":{
        "s3SchemaVersion":"1.0",
        "configurationId":"testConfigRule",
        "bucket":{
          "name":"amzn-s3-demo-bucket",
          "ownerIdentity":{
            "principalId":"A3NL1K0ZZKExample"
          },
          "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
        },
        "object":{
          "key":"expiration/delete",
          "sequencer":"0055AED6DCD90281E5",
        }
      }
    }
  ]
}
```

Nachrichten, die Amazon S3 sendet, um ein `s3:LifecycleTransition` Ereignis zu veröffentlichen, enthalten auch die folgenden Informationen.

```
"lifecycleEventData":{
  "transitionEventData": {
    "destinationStorageClass": the destination storage class for the object
  }
}
```

Elemente der Lebenszyklus-Konfiguration

Themen

- [ID-Element](#)
- [Statuselement](#)
- [Filterelement](#)
- [Elemente, die Lebenszyklus-Aktionen beschreiben](#)

Sie geben eine Amazon S3 S3-Lifecycle-Konfiguration anXML, die aus einer oder mehreren Lifecycle-Regeln besteht.

```
<LifecycleConfiguration>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
</LifecycleConfiguration>
```

Jeder Regel umfasst Folgendes:

- Regelmetadaten, die eine Regel-ID und einen Status enthalten, der angibt, ob die Regel aktiviert oder deaktiviert ist. Wenn eine Regel deaktiviert ist, führt Amazon S3 keine in der Regel spezifizierten Aktionen aus.
- Ein Filter, der die Objekte identifiziert, für die die Regel gilt. Sie können einen Filter mithilfe der Objektgröße, des Objektschlüsselpräfixes, eines oder mehrerer Objekt-Tags oder einer Kombination von Filtern angeben.

- Eine oder mehrere Übergangs- oder Ablaufaktionen mit einem Datum oder einem Zeitintervall innerhalb der Lebensdauer des Objekts, zu denen Amazon S3 die angegebene Aktion ausführen soll.

In den folgenden Abschnitten werden die XML Elemente einer S3-Lifecycle-Konfiguration beschrieben. Beispielkonfigurationen finden Sie unter [Beispiele der S3-Lebenszyklus-Konfiguration](#).

ID-Element

Eine S3-Lebenszyklus-Konfiguration kann bis zu 1.000 Regeln haben. Diese Grenze ist nicht einstellbar. Das <ID> Element identifiziert eine Regel eindeutig. Die Länge der ID ist auf 255 Zeichen begrenzt.

Statuselement

Der <Status> Elementwert kann entweder Enabled oder seinDisabled. Wenn eine Regel deaktiviert ist, führt Amazon S3 keine in der Regel definierten Aktionen aus.

Filterelement

Eine Lebenszyklusregel kann auf Grundlage des <Filter> Elements, das Sie in der Lebenszyklusregel angeben, für alle oder eine Teilmenge von Objekten in einem Bucket gelten.

Sie können Objekte nach dem Schlüsselpräfix, Objekt-Markierungen oder Kombinationen aus beidem filtern (bei einer Kombination verwendet Amazon S3 ein logisches AND für die Filter). Betrachten Sie die folgenden Beispiele:

- Angeben eines Filters mithilfe von Schlüsselpräfixen — Dieses Beispiel zeigt eine S3-Lebenszyklusregel, die für eine Teilmenge von Objekten gilt, die auf dem Schlüsselnamenpräfix () basiert. logs/ Die Lebenszyklusregel gilt beispielsweise für die Objekte logs/mylog.txtlogs/temp1.txt, und. logs/test.txt Die Regel gilt nicht für das Objekt example.jpg.

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    transition/expiration actions
    ...
  </Rule>
  ...
```

```
</LifecycleConfiguration>
```

Wenn Sie eine Lebenszyklusaktion auf eine Teilmenge von Objekten anwenden möchten, die auf unterschiedlichen Schlüsselnamenpräfixen basieren, geben Sie separate Regeln an. Geben Sie in jeder Regel einen auf einem Präfix basierenden Filter an. Um beispielsweise eine Lebenszyklusaktion für Objekte mit den Schlüsselpräfixen `projectA/` und zu beschreiben `projectB/`, geben Sie zwei Regeln wie folgt an:

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Prefix>projectA</Prefix>
    </Filter>
    transition/expiration actions
    ...
  </Rule>

  <Rule>
    <Filter>
      <Prefix>projectB</Prefix>
    </Filter>
    transition/expiration actions
    ...
  </Rule>
</LifecycleConfiguration>
```

Weitere Informationen über Objektschlüssel finden Sie unter [Erstellen von Objektschlüsselnamen](#).

- Angeben eines Filters auf der Grundlage von Objekt-Tags — Im folgenden Beispiel gibt die Lebenszyklusregel einen Filter an, der auf einem Tag (*key*) und einem Wert (*value*) basiert. Die Regel wird dann nur auf eine Untermenge von Objekten mit dem spezifischen Tag angewendet.

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Tag>
        <Key>key</Key>
        <Value>value</Value>
      </Tag>
    </Filter>
    transition/expiration actions
    ...
  </Rule>
</LifecycleConfiguration>
```

```
</Rule>
</LifecycleConfiguration>
```

Sie können einen Filter auf mehreren Markierungen basierend angeben. Sie müssen die Tags in das `<And>` Element einschließen, wie im folgenden Beispiel gezeigt. Die Regel weist Amazon S3 an, Lebenszyklus-Aktionen für Objekte mit zwei Markierungen auszuführen (mit dem spezifischen Tag-Schlüssel und -wert).

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <And>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key2</Key>
          <Value>value2</Value>
        </Tag>
        ...
      </And>
    </Filter>
    transition/expiration actions
  </Rule>
</Lifecycle>
```

Die Lebenszyklusregel gilt für Objekte, für die beide Markierungen angegeben sind. Amazon S3 führt eine logische AND-Operation aus. Beachten Sie Folgendes:

- Jedes Tag muss sowohl dem Schlüssel als auch dem Wert exakt entsprechen. Wenn Sie nur ein `<Key>` Element und kein `<Value>` Element angeben, gilt die Regel nur für Objekte, die dem Tag-Schlüssel entsprechen und für die kein Wert angegeben ist.
- Die Regel gilt für die Untermenge der Objekte, die alle in der Regel angegebenen Markierungen besitzt. Wenn für ein Objekt zusätzliche Markierungen angegeben sind, gilt die Regel weiterhin.

Note

Wenn Sie mehrere Markierungen in einem Filter spezifizieren, muss jeder Tag-Schlüssel eindeutig sein.

- Angeben eines Filters, der sowohl auf dem Präfix als auch auf einem oder mehreren Tags basiert — In einer Lebenszyklusregel können Sie einen Filter angeben, der sowohl auf dem key prefix als auch auf einem oder mehreren Tags basiert. Auch hier müssen Sie alle diese Filterelemente wie folgt in das <And> Element einschließen:

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <And>
        <Prefix>key-prefix</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key2</Key>
          <Value>value2</Value>
        </Tag>
        ...
      </And>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions
  </Rule>
</LifecycleConfiguration>
```

Amazon S3 kombiniert diese Filter mithilfe eines logischen AND. Das heißt, die Regel gilt für die Teilmenge der Objekte mit dem angegebenen key prefix und den angegebenen Tags. Ein Filter kann höchstens ein Präfix und null oder mehr Markierungen aufweisen.

- Sie können einen leeren Filter angeben, dann gilt die Regel für alle Objekte in dem Bucket.

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions
  </Rule>
</LifecycleConfiguration>
```


- Um eine Regel nach Objektgröße zu filtern, können Sie eine Mindestgröße (`ObjectSizeGreaterThan`) oder eine Höchstgröße (`ObjectSizeLessThan`) oder einen Bereich von Objektgrößen angeben.

Objektgrößenwerte sind in Bytes. Standardmäßig werden Objekte, die kleiner als 128 KB sind, nicht in eine Speicherklasse übertragen, es sei denn, Sie geben eine kleinere Mindestgröße (`ObjectSizeGreaterThan`) oder eine Maximalgröße (`ObjectSizeLessThan`) an. Weitere Informationen finden Sie unter [Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind](#).

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
    </Filter>
    <Status>Enabled</Status>
    transition/expiration actions
  </Rule>
</LifecycleConfiguration>
```

Note

Die `ObjectSizeLessThan` Filter `ObjectSizeGreaterThan` und schließen die angegebenen Werte aus. Wenn Sie beispielsweise Objekte mit einer Größe von 128 KB bis 1024 KB so einstellen, dass sie von der Speicherklasse S3 Standard zur Speicherklasse S3 Standard-IA verschoben werden, werden Objekte, die genau 1024 KB und 128 KB groß sind, nicht in S3 Standard-IA umgewandelt. Stattdessen gilt die Regel nur für Objekte, die größer als 128 KB und kleiner als 1024 KB sind.

Wenn Sie einen Objektgrößenbereich angeben, muss die `ObjectSizeGreaterThan`-Ganzzahl kleiner als der `ObjectSizeLessThan`-Wert sein. Wenn Sie mehr als einen Filter verwenden, müssen Sie die Filter in ein `<And>`-Element packen. Das folgende Beispiel zeigt, wie Objekte in einem Bereich zwischen 500 Byte und 64.000 Byte angegeben werden.

```
<LifecycleConfiguration>
  <Rule>
```

```
<Filter>
  <And>
    <Prefix>key-prefix</Prefix>
    <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
    <ObjectSizeLessThan>64000</ObjectSizeLessThan>
  </And>
</Filter>
<Status>Enabled</Status>
  transition/expiration actions
</Rule>
</LifecycleConfiguration>
```

Elemente, die Lebenszyklus-Aktionen beschreiben

Sie können Amazon S3 anweisen, spezifische Aktionen innerhalb der Lebensdauer eines Objekts auszuführen, indem Sie eine oder mehrere vordefinierte Aktionen in einer S3-Lebenszyklusregel angeben. Die Wirkung dieser Aktionen ist vom Versioning-Status Ihres Buckets abhängig.

- **Transition**Aktionselement — Sie geben die Transition Aktion an, um Objekte von einer Speicherklasse in eine andere zu übertragen. Weitere Informationen zum Übergang von Objekten finden Sie unter [Unterstützte Übergänge](#). Wenn ein vorgegebenes Datum oder einem Zeitintervall innerhalb der Lebensdauer des Objekts erreicht ist, führt Amazon S3 den Übergang aus.

Für einen versionsfähigen Bucket (Bucket mit aktiviertem oder ausgesetztem Versioning) wird die Transition-Aktion auf die aktuelle Objektversion angewendet. Um nicht aktuelle Versionen zu verwalten, definiert Amazon S3 die NoncurrentVersionTransition-Aktion (wird später in diesem Thema beschrieben).

- **Expiration**Aktionselement — Die Expiration Aktion lässt die in der Regel identifizierten Objekte ablaufen und gilt für geeignete Objekte in einer der Amazon S3 S3-Speicherklassen. Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#). Amazon S3 lässt keinen Zugriff auf abgelaufene Objekte zu. Ob die Objekte dauerhaft entfernt werden, ist vom Versioning-Status des Buckets abhängig.
 - Bucket ohne Versionsnummer — Die Expiration Aktion führt dazu, dass Amazon S3 das Objekt dauerhaft entfernt.
 - Versionsfähiger Bucket – Für einen versionsfähigen Bucket (d. h. das Versioning ist aktiviert oder ausgesetzt) gibt es mehrere Aspekte, die bestimmen, wie Amazon S3 Expiration-Aktion verarbeitet. Für Buckets mit aktivierter oder ausgesetzter Versionsverwaltung gilt Folgendes:

- Die `Expiration`-Aktion wird nur auf die aktuelle Version angewendet (sie wirkt sich nicht auf nicht aktuelle Objektversionen aus).
- Amazon S3 führt keine Aktion aus, wenn es eine oder mehrere Objektversionen gibt und die Löschkennzeichnung die aktuelle Version ist.
- Wenn die aktuelle Objektversion die einzige Objektversion und auch eine Löschkennzeichnung ist (auch als Löschkennzeichnung eines abgelaufenen Objekts bezeichnet, wobei alle Objektversionen gelöscht werden und nur noch eine Löschkennzeichnung beibehalten wird), entfernt Amazon S3 die Löschkennzeichnung des abgelaufenen Objekts. Sie können die Ablaufaktion auch verwenden, um Amazon S3 anzuweisen, alle abgelaufenen Löschkennzeichnungen zu entfernen. Ein Beispiel finden Sie unter [Beispiel 7: Löschen abgelaufener Löschkennzeichnungen für Objekte](#).

Weitere Informationen finden Sie unter [Verwenden der Versionierung in S3-Buckets](#).

Berücksichtigen Sie auch die folgenden Punkte, wenn Sie Amazon S3 zur Ablaufverwaltung einrichten:

- Bucket mit aktiviertem Versionierung

Wenn die aktuelle Objektversion keine Löschkennzeichnung ist, fügt Amazon S3 die Löschkennzeichnung mit einer eindeutigen Versions-ID hinzu. Damit ist die aktuelle Version nicht mehr aktuell und die Löschkennzeichnung wird zur aktuellen Version.

- Bucket mit ausgesetztem Versionierung

In einem Bucket, für das die Versionierung gesperrt wurde, veranlasst die Ablaufaktion, dass Amazon S3 eine Löschkennzeichnung mit der `null` Versions-ID erstellt. Diese Löschkennzeichnung ersetzt jede Objektversion mit einer Versions-ID von `null` in der Versionshierarchie, womit das Objekt effektiv gelöscht wird.

Darüber hinaus unterstützt Amazon S3 die folgenden Aktionen, mit denen Sie nicht aktuelle Objektversionen in einem versionsfähigen Bucket verwalten können (d. h. für Buckets mit aktiviertem und ausgesetztem Versionierung).

- **NoncurrentVersionTransition**Aktionselement — Verwenden Sie diese Aktion, um anzugeben, wann Amazon S3 Objekte in die angegebene Speicherklasse überführt. Sie können diesem Ablauf eine bestimmte Anzahl von Tagen zugrunde legen, seit die Objekte nicht mehr aktuell sind. Zusätzlich zur Anzahl der Tage können Sie auch eine maximale Anzahl an veralteten Versionen angeben, die aufbewahrt werden sollen (zwischen 1 und 100). Dieser Wert bestimmt,

wie viele neuere, nicht aktuelle Versionen vorhanden sein müssen, bevor Amazon S3 die zugehörige Aktion für eine bestimmte Version ausführen kann. Amazon S3 stellt alle zusätzlichen Langzeitversionen um, die über die angegebene Anzahl hinaus beibehalten werden sollen.

Um die maximale Anzahl an nicht aktuellen Versionen anzugeben, müssen Sie auch ein `Filter` Element angeben. Wenn Sie kein `Filter` Element angeben, generiert Amazon S3 einen `InvalidRequest` Fehler, wenn Sie eine maximale Anzahl von nicht aktuellen Versionen angeben.

Weitere Informationen zum Übergang von Objekten finden Sie unter [Unterstützte Übergänge](#). Ausführliche Informationen dazu, wie Amazon S3 das Datum berechnet, wenn Sie die Anzahl der Tage in der `NoncurrentVersionTransition`-Aktion angeben, finden Sie unter [Lebenszyklusregeln: Basierend auf dem Alter eines Objekts](#).

- **`NoncurrentVersionExpiration`**Aktionselement — Verwenden Sie diese Aktion, um Amazon S3 anzuweisen, nicht aktuelle Versionen von Objekten dauerhaft zu löschen. Diese gelöschten Objekte können nicht wiederhergestellt werden. Sie können dieses Ablaufdatum auf eine bestimmte Anzahl von Tagen beziehen, seit die Objekte nicht mehr aktuell sind. Zusätzlich zur Anzahl der Tage können Sie auch eine maximale Anzahl an veralteten Versionen angeben, die aufbewahrt werden sollen (zwischen 1 und 100). Dieser Wert gibt die Anzahl der neueren nicht aktuellen Versionen an, die vorhanden sein müssen, damit Amazon S3 die zugehörige Aktion für eine bestimmte Version ausführen kann. Amazon S3 löscht alle zusätzlichen älteren Versionen, die über die angegebene Anzahl hinaus aufbewahrt werden sollen, dauerhaft.

Um die maximale Anzahl an nicht aktuellen Versionen anzugeben, müssen Sie auch ein `Filter` Element angeben. Wenn Sie kein `Filter` Element angeben, generiert Amazon S3 einen `InvalidRequest` Fehler, wenn Sie eine maximale Anzahl von nicht aktuellen Versionen angeben.

Das verzögerte Entfernen nicht aktueller Objekte kann hilfreich sein, wenn Sie versehentliche Lösch- oder Überschreibvorgänge korrigieren müssen. Beispielsweise können Sie eine Ablaufregel konfigurieren, um nicht aktuelle Versionen fünf Tage nach dem Zeitpunkt zu löschen, zu dem sie nicht aktuell geworden sind. Nehmen wir zum Beispiel an, dass Sie am 1.1.2014 um 10:30 Uhr UTC ein Objekt mit dem Namen `photo.gif` (Versions-ID 111111) erstellen. Am 02.01.2014 um 11:30 Uhr löschen Sie versehentlich `photo.gif` (Versions-ID 111111)UTC, wodurch eine Löschmarke mit einer neuen Versions-ID (z. B. Versions-ID 4857693) erstellt wird. Jetzt haben Sie fünf Tage Zeit, die Originalversion von `photo.gif` (Versions-ID 111111) wiederherzustellen, bevor das Löschen permanent wird. Am 08.01.2014 um 00:00 UTC Uhr wird die Lebenszyklusregel

für den Ablauf ausgeführt und dauerhaft gelöscht `photo.gif` (Versions-ID 111111), fünf Tage, nachdem sie zu einer nicht aktuellen Version geworden ist.

Weitere Informationen darüber, wie Amazon S3 das Datum berechnet, an dem Sie die Anzahl der Tage `NoncurrentVersionExpiration` Aktion angeben finden Sie in [Lebenszyklusregeln: Basierend auf dem Alter eines Objekts](#).

Note

Bei den Lebenszykluskonfigurationen für den Ablauf von Objekten werden unvollständige mehrteilige Uploads nicht entfernt. Um unvollständige mehrteilige Uploads zu entfernen, müssen Sie die Aktion „`AbortIncompleteMultipartUpload` Lebenszyklus-Konfiguration“ verwenden, die weiter unten in diesem Abschnitt beschrieben wird.

Zusätzlich zu den Übergangs- und Ablaufaktionen können Sie die folgenden Lifecycle-Konfigurationsaktionen verwenden, um Amazon S3 anzuweisen, unvollständige mehrteilige Uploads zu beenden oder Markierungen für das Löschen abgelaufener Objekte zu entfernen:

- **`AbortIncompleteMultipartUpload`**Aktionselement — Verwenden Sie dieses Element, um eine maximale Dauer (in Tagen) festzulegen, für die mehrteilige Uploads in Bearbeitung bleiben sollen. Wenn die entsprechenden mehrteiligen Uploads (bestimmt durch den in der Lebenszyklusregel `prefix` angegebenen Schlüsselnamen) nicht innerhalb des vordefinierten Zeitraums erfolgreich abgeschlossen werden, stoppt Amazon S3 die unvollständigen mehrteiligen Uploads. Weitere Informationen finden Sie unter [Abbrechen eines mehrteiligen Uploads](#).

Note

Sie können diese Lebenszyklusaktion nicht in einer Regel angeben, die über einen Filter verfügt, der Objekt-Tags verwendet.

- **`ExpiredObjectDeleteMarker`**Aktionselement — In einem Bucket mit aktivierter Versionierung wird eine Löschmarkierung ohne ältere Versionen als Löschmarkierung für abgelaufene Objekte bezeichnet. Sie können diese Lebenszyklusaktion verwenden, um Amazon S3 anzuweisen, Markierungen für das Löschen abgelaufener Objekte zu entfernen. Ein Beispiel finden Sie unter [Beispiel 7: Löschen abgelaufener Löschmarkierungen für Objekte](#).

Note

Sie können diese Lebenszyklusaktion nicht in einer Regel angeben, die über einen Filter verfügt, der Objekt-Tags verwendet.

Wie Amazon S3 berechnet, wie lange ein Objekt nicht aktuell ist

In einem versionsfähigen Bucket können Sie mehrere Versionen eines Objekts haben. Es gibt immer eine aktuelle Version und null oder mehr nicht aktuelle Versionen. Immer wenn Sie ein Objekt hochladen, wird die aktuelle Version als die nicht aktuelle Version beibehalten, und die neu hinzugefügte Version, der Nachfolger, wird zur aktuellen Version. Um die Anzahl der Tage zu bestimmen, wie lange ein Objekt nicht aktuell ist, wertet Amazon S3 es aus, wenn der Nachfolger erstellt wurde. Amazon S3 verwendet die Anzahl der Tage, seit der Nachfolger erstellt wurde, als die Anzahl der Tage, wie lange ein Objekt nicht aktuell ist.

Wiederherstellung vorheriger Versionen eines Objekts bei Verwendung von S3-Lebenszyklus-Konfigurationen

Wie unter erklärt [Wiederherstellen früherer Versionen](#), können Sie eine der folgenden beiden Methoden verwenden, um frühere Versionen eines Objekts abzurufen:

- Methode 1 — Kopieren Sie eine nicht aktuelle Version des Objekts in denselben Bucket. Das kopierte Objekt wird zur aktuellen Version dieses Objekts, und alle Objektversionen werden beibehalten.
- Methode 2 — Die aktuelle Version des Objekts dauerhaft löschen. Wenn Sie die aktuelle Objektversion löschen, wandeln Sie letztlich die nicht aktuelle Version in die aktuelle Version dieses Objekts um.

Wenn Sie S3 Lifecycle-Konfigurationsregeln mit Buckets mit aktivierter Versionierung verwenden, empfehlen wir als bewährte Methode, Methode 1 zu verwenden.

Der S3-Lebenszyklus arbeitet nach einem letztlich konsistenten Modell. Eine aktuelle Version, die Sie dauerhaft gelöscht haben, verschwindet möglicherweise erst, wenn die Änderungen auf alle Amazon S3-Systeme übertragen wurden. (Daher weiß Amazon S3 möglicherweise vorübergehend nichts von dieser Löschung.) In der Zwischenzeit kann die Lebenszyklusregel, die Sie für den Ablauf nicht aktueller Objekte konfiguriert haben, die nicht aktuellen Objekte dauerhaft entfernen, auch dasjenige, das Sie wiederherstellen möchten.

Daher ist das Kopieren der alten Version, wie in Methode 1 empfohlen, die sicherere Alternative.

Lebenszyklus-Aktionen und der Versioning-Status eines Buckets

Lebenszyklusregeln: Basierend auf dem Alter eines Objekts

Sie können einen Zeitraum in der Anzahl der Tage ab der Erstellung (oder Änderung) des Objekts angeben, in dem Amazon S3 die angegebene Aktion ausführen kann.

Wenn Sie die Anzahl der Tage in den `Transition`- und `Expiration`-Aktionen in einer S3-Lebenszyklus-Konfiguration angeben, beachten Sie Folgendes:

- Der Wert, den Sie angeben, ist die Anzahl der Tage seit der Objekterstellung, an denen die Aktion ausgeführt wird.
- Amazon S3 berechnet die Zeit, indem es die in der Regel angegebene Anzahl von Tagen zur Objekterstellungszeit addiert und die resultierende Zeit auf den nächsten Tag um Mitternacht rundet. UTC Wenn ein Objekt beispielsweise am 15.01.2014 um 10:30 Uhr erstellt wurde UTC und Sie in einer Übergangsregel 3 Tage angeben, würde das Übergangsdatum des Objekts als 19.01.2014 00:00 Uhr berechnet. UTC

Note

Amazon S3 behält nur das letzte Änderungsdatum für jedes Objekt bei. Beispielsweise zeigt die Amazon S3 S3-Konsole das Datum der letzten Änderung im Eigenschaftenbereich des Objekts an. Wenn Sie zum ersten Mal ein neues Objekt erstellen, spiegelt dieses Datum das Datum wider, an dem das Objekt erstellt wurde. Wenn Sie das Objekt ersetzen, ändert sich das Datum entsprechend. Daher ist das Erstellungsdatum gleichbedeutend mit dem Datum der letzten Änderung.

Wenn Sie die Anzahl der Tage in den `NoncurrentVersionTransition`- und `NoncurrentVersionExpiration`-Aktionen in einer Lebenszyklus-Konfiguration angeben, beachten Sie Folgendes:

- Der Wert, den Sie angeben, ist die Anzahl der Tage, ab dem die Version des Objekts veraltet ist (d. h. wenn das Objekt überschrieben oder gelöscht wird), an denen Amazon S3 die Aktion für das oder die angegebenen Objekte ausführt.
- Amazon S3 berechnet die Zeit, indem es die in der Regel angegebene Anzahl von Tagen zu dem Zeitpunkt addiert, zu dem die neue Nachfolgerversion des Objekts erstellt wird, und die resultierende Zeit auf den nächsten Tag um Mitternacht rundet. UTC Nehmen wir zum Beispiel an, dass Sie in Ihrem Bucket eine aktuelle Version eines Objekts haben, das am 1.1.2014 um 10:30 Uhr erstellt wurde. UTC Wenn die neue Version des Objekts, das die aktuelle Version ersetzt, am 15.01.2014 um 10:30 Uhr erstellt wird und Sie in einer Übergangsregel 3 Tage angeben UTC, wird das Übergangsdatum des Objekts als 19.01.2014 00:00 Uhr berechnet. UTC

Lebenszyklusregeln: Basierend auf einem spezifischen Datum

Wenn Sie in einer S3-Lebenszyklus-Regel eine Aktion angeben, können Sie ein Datum angeben, wann Amazon S3 die Aktion ausführen soll. Wenn das spezifische Datum erreicht ist, wendet Amazon S3 die Aktion auf alle qualifizierten Objekte an (basierend auf den Filterkriterien).

Wenn Sie eine S3-Lebenszyklusaktion mit einem Datum angeben, das in der Vergangenheit liegt, kommen alle qualifizierten Objekte sofort für diese Lebenszyklusaktion in Frage.

Important

Die auf einem Datum basierende Aktion ist keine einmalige Aktion. Amazon S3 wendet die auf dem Datum basierende Aktion auch an, nachdem das Datum erreicht wurde, solange der Regel-Status is Enabled.

Nehmen wir beispielsweise an, dass Sie eine datumsbasierte Expiration Aktion angeben, um alle Objekte zu löschen (gehen Sie davon aus, dass in der Regel kein Filter angegeben ist). Amazon S3 lässt zu dem angegebenen Datum alle Objekte in dem Bucket ablaufen. Amazon S3 läuft auch weiterhin für alle neuen Objekte ab, die Sie im Bucket erstellen. Um die Lebenszyklusaktion zu beenden, müssen Sie entweder die Aktion aus der Lebenszyklusregel entfernen, die Regel deaktivieren oder die Regel aus der Lebenszykluskonfiguration löschen.

Der Datumswert muss dem Format ISO 8601 entsprechen. Die Zeit ist immer Mitternacht. UTC

Note

Sie können mit der Amazon S3 S3-Konsole keine datumsbasierten Lebenszyklusregeln erstellen, aber Sie können solche Regeln anzeigen, deaktivieren oder löschen.

Beispiele der S3-Lebenszyklus-Konfiguration

Dieser Abschnitt enthält Beispiele für S3-Lebenszyklus-Konfigurationen. Jedes Beispiel zeigt, wie Sie das XML in jedem der Beispielszenarien spezifizieren können.

Themen

- [Beispiel 1: Festlegen eines Filters](#)
- [Beispiel 2: Deaktivieren einer Lebenszyklusregel](#)
- [Beispiel 3: Verkleinerung der Speicherklasse während der Lebensdauer eines Objekts](#)
- [Beispiel 4: Festlegen mehrerer Regeln](#)
- [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht](#)
- [Beispiel 6: Spezifikation einer Lebenszyklus-Konfigurationsregel für einen Bucket mit Versioning](#)
- [Beispiel 7: Löschen abgelaufener Löschmarkierungen für Objekte](#)
- [Beispiel 8: Lebenszyklus-Konfigurationsregel für das Abbrechen mehrteiliger Uploads](#)
- [Beispiel 9: Lebenszyklus-Konfiguration mit größenbasierten Regeln](#)
- [Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind](#)

Beispiel 1: Festlegen eines Filters

Jede S3-Lebenszyklusregel enthält einen Filter, mit dem Sie eine Untermenge der Objekte in Ihrem Bucket identifizieren können, auf die sich die S3-Lebenszyklusregel bezieht. Das folgenden S3 Lifecycle-Konfigurationen zeigen Beispiele dafür, wie Sie einen Filter spezifizieren können.

- In dieser S3-Lebenszyklus-Konfigurationregel spezifiziert der Filter ein Schlüsselpräfix (`tax/`). Aus diesem Grund gilt die Regel für Objekte mit dem Schlüsselnamenpräfix `tax/`, wie beispielsweise `tax/doc1.txt` und `tax/doc2.txt`.

Die Regel spezifiziert zwei Aktionen, die Amazon S3 zu Folgendem anweisen:

- Übergang von Objekten in die Speicherklasse S3 Glacier Flexible Retrieval 365 Tage (ein Jahr) nach der Erstellung.
- Objekte 3.650 Tage (10 Jahre) nach der Erstellung löschen (die Expiration-Aktion).

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition and Expiration Rule</ID>
    <Filter>
      <Prefix>tax/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Anstatt das Objektalter in Tagen nach der Erstellung anzugeben, können Sie für jede Aktion ein Datum angeben. Sie können Date und Days nicht in derselben Regel kombinieren.

- Wenn Sie wollen, dass die S3-Lebenszyklusregel für alle Objekte im Bucket gilt, geben Sie ein leeres Präfix an. In der folgenden Konfiguration gibt die Regel eine Transition-Aktion an, die Amazon S3 anweist, Objekte 0 Tage nach der Erstellung in die S3 Glacier Flexible Retrieval-Speicherklasse zu überführen. Diese Regel bedeutet, dass die Objekte um Mitternacht UTC nach der Erstellung in S3 Glacier Flexible Retrieval archiviert werden können. Weitere Informationen zu Lebenszykluseinschränkungen finden Sie unter [Einschränkungen und Überlegungen für Übergänge](#).

```
<LifecycleConfiguration>
  <Rule>
    <ID>Archive all object same-day upon creation</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
```

```

    <StorageClass>GLACIER</StorageClass>
  </Transition>
</Rule>
</LifecycleConfiguration>

```

- Sie können null oder mehrere Schlüsselnamenpräfixe und null oder mehr Objekt-Markierungen in einem Filter angeben. Der folgende Beispiel-Code wendet die S3-Lebenszyklusregel auf eine Untermenge von Objekten mit dem Schlüsselpräfix `tax/` an, ebenso wie auf Objekte mit zwei Markierungen mit spezifischem Schlüssel und Wert. Wenn Sie mehr als einen Filter angeben, müssen Sie das `<And>`-Element wie gezeigt einschließen (Amazon S3 wendet ein logisches AND an, um die angegebenen Filterbedingungen zu kombinieren).

```

...
<Filter>
  <And>
    <Prefix>tax/</Prefix>
    <Tag>
      <Key>key1</Key>
      <Value>value1</Value>
    </Tag>
    <Tag>
      <Key>key2</Key>
      <Value>value2</Value>
    </Tag>
  </And>
</Filter>
...

```

- Sie können Objekte nur auf Markierungen basierend filtern. Die folgende S3-Lebenszyklusregel beispielsweise wird auf Objekte angewendet, die die beiden spezifizierten Markierungen aufweisen (es wird kein Präfix angegeben).

```

...
<Filter>
  <And>
    <Tag>
      <Key>key1</Key>
      <Value>value1</Value>
    </Tag>
    <Tag>
      <Key>key2</Key>
    </Tag>
  </And>
</Filter>
...

```

```
        <Value>value2</Value>
      </Tag>
    </And>
  </Filter>
  ...
```

Important

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von [Löschmarkierungen](#).
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht](#).

Beispiel 2: Deaktivieren einer Lebenszyklusregel

Sie können eine S3-Lebenszyklusregel vorübergehend deaktivieren. Die folgende S3-Lebenszyklus-Konfiguration spezifiziert zwei Regeln:

- Regel 1 weist Amazon S3 an, Objekte mit dem Präfix `logs/` bald nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen.
- Regel 2 weist Amazon S3 an, Objekte mit dem Präfix `documents/` bald nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen.

In der Konfiguration ist Regel 1 aktiviert und Regel 2 ist deaktiviert. Amazon S3 ignoriert deaktivierte Regeln.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule2</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Disabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Beispiel 3: Verkleinerung der Speicherklasse während der Lebensdauer eines Objekts

In diesem Beispiel nutzen Sie die S3-Lebenszyklus-Konfiguration, um die Speicherklasse von Objekten über ihre Lebensdauer stufenweise zu reduzieren. Diese schichtweise Reduzierung kann dazu beitragen, Speicherkosten zu reduzieren. Weitere Informationen zu Preisen finden Sie unter [Amazon-S3-Preise](#).

Die folgende S3-Lebenszyklus-Konfiguration spezifiziert eine Regel, die auf Objekte mit Schlüsselnamenpräfix `logs/` angewendet wird. Die Regel definiert die folgenden Aktionen:

- Zwei Übergangskaktionen:
 - Übergang von Objekten in die Speicherklasse S3 Standard-IA 30 Tage nach der Erstellung.
 - Übergang von Objekten in die Speicherklasse S3 Glacier Flexible Retrieval 90 Tage nach der Erstellung.
- Eine Ablaufaktion, die Amazon S3 anweist, Objekte ein Jahr nach ihrer Erstellung zu löschen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>example-id</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>90</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Note

Sie können eine Regel verwenden, um alle S3-Lebenszyklus-Aktionen zu beschreiben, die für dieselbe Objektmenge angewendet werden (identifiziert durch den Filter). Andernfalls können Sie mehrere Regeln hinzufügen, die jeweils einen unterschiedlichen Filter angeben.

Important

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt für mehrere S3 Lifecycle-Aktionen am selben Tag in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von [Löschmarkierungen](#).
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht.](#)

Beispiel 4: Festlegen mehrerer Regeln

Sie können mehrere Regeln angeben, wenn Sie unterschiedliche S3-Lebenszyklus-Aktionen auf unterschiedliche Objekte anwenden wollen. Die folgende S3-Lebenszyklus-Konfiguration spezifiziert zwei Regeln:

- Regel 1 gilt für Objekte mit dem Schlüsselnamenpräfix `classA/`. Sie weist Amazon S3 an, Objekte ein Jahr nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen, und diese Objekte 10 Jahre nach dem Erstellen ablaufen zu lassen.
- Regel 2 gilt für Objekte mit dem Schlüsselnamenpräfix `classB/`. Sie weist Amazon S3 an, Objekte 90 Tage nach der Erstellung in die Speicherklasse S3 Standard-IA zu übertragen, und sie ein Jahr nach dem zu löschen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>ClassADocRule</ID>
    <Filter>
      <Prefix>classA</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>ClassBDocRule</ID>
    <Filter>
      <Prefix>classB</Prefix>
    </Filter>
    <Status>Enabled</Status>
```

```
<Transition>
  <Days>90</Days>
  <StorageClass>STANDARD_IA</StorageClass>
</Transition>
<Expiration>
  <Days>365</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

Important

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von [Löschmarkierungen](#).
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht](#).

Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht

Sie könnten eine S3-Lebenszyklus-Konfiguration angeben, in der Sie überlappende Präfixe oder Aktionen spezifizieren.

Im Allgemeinen gibt S3-Lebenszyklus Kostenoptimierung Vorrang. Wenn sich z. B. zwei Ablaufrichtlinien überschneiden, wird die Ablaufrichtlinie mit der kürzeren Frist durchgesetzt, sodass die Daten nicht länger als erwartet gespeichert werden. Wenn sich zwei Übergangsrichtlinien überschneiden, überführt S3 Lifecycle Ihre Objekte in die Speicherklasse mit geringeren Kosten.

In beiden Fällen versucht S3 Lifecycle den für Sie kostengünstigsten Pfad auszuwählen. Die Speicherklasse S3 Intelligent-Tiering ist von dieser Regel ausgenommen. S3 Intelligent-Tiering wird

von S3 Lifecycle gegenüber jeder Speicherklasse bevorzugt, abgesehen von den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive.

Die folgenden Beispiele zeigen, wie Amazon S3 potenzielle Konflikte löst.

Example 1: Überlappende Präfixe (kein Konflikt)

Die folgende Beispielkonfiguration weist zwei Regeln auf, die überlappenden Präfixe spezifizieren, wie folgt:

- Die erste Regel spezifiziert einen leeren Filter, d. h. alle Objekte in dem Bucket werden angesprochen.
- Die zweite Regel spezifiziert ein Schlüsselnamenpräfix (logs/), d. h. nur eine Untermenge von Objekten.

Regel 1 fordert Amazon S3 auf, alle Objekte ein Jahr nach der Erstellung zu löschen. Regel 2 fordert Amazon S3 auf, 30 Tage nach der Erstellung eine Teilmenge von Objekten in die S3-Standard-IA-Speicherklasse zu überführen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
      <Days>30</Days>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Da in diesem Fall kein Konflikt besteht, überführt Amazon S3 die Objekte mit dem Präfix `logs/` 30 Tage nach der Erstellung in die Speicherklasse S3 Standard-IA. Wenn ein Objekt ein Jahr nach der Erstellung erreicht, wird es gelöscht.

Example 2: Widersprüchliche Lebenszyklus-Aktionen

In dieser Beispielkonfiguration gibt es zwei Regeln, die Amazon S3 anweisen, zwei unterschiedliche Aktionen für dieselbe Objektmenge zur selben Zeit in der Lebensdauer des Objekts auszuführen:

- Beide Regeln geben dasselbe Schlüsselnamenpräfix an, deshalb gelten beide Regeln für dieselbe Objektmenge.
- Beide Regeln spezifizieren dieselben 365 Tage nach der Erstellung des Objekts, wann die Regeln angewendet werden sollen.
- Eine Regel weist Amazon S3 an, Objekte zur S3 Standard-IA-Speicherklasse zu überführen. Eine andere Regel weist Amazon S3 an, die Objekte zur gleichen Zeit ablaufen zu lassen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

In diesem Fall wollen Sie, dass Objekte ablaufen (zu entfernen), deshalb macht es keinen Sinn, die Speicherklasse zu ändern, und Amazon S3 wählt einfach die Ablaufaktion für diese Objekte aus.

Example 3: Überlappende Präfixe, die zu widersprüchlichen Lebenszyklus-Aktionen führen

In diesem Beispiel besitzt die Konfiguration zwei Regeln, die überlappende Präfixe angeben, wie folgt:

- Regel 1 legt ein leeres Präfix fest (was für alle Objekte gilt).
- Regel 2 spezifiziert ein Schlüsselnamenpräfix (logs/), das eine Untermenge aller Objekte angibt.

Für die Untermenge der Objekte mit dem Schlüsselnamenpräfix logs/ werden die S3-Lebenszyklus-Aktionen aus beiden Regeln angewendet. Eine Regel weist Amazon S3 an, Objekte 10 Tage nach der Erstellung zu übertragen, und eine andere Regel weist Amazon S3 an, Objekte 365 Tage nach dem Erstellen zu übertragen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
      <Days>10</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA</StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

In diesem Fall entscheidet Amazon S3 den Übergang 10 Tage nach der Erstellung auszuführen.

Example 4: Tag-basiertes Filtern, das zu widersprüchlichen Lebenszyklus-Aktionen führt

Angenommen, Sie haben die folgende S3-Lebenszyklus-Konfiguration, die zwei Regeln enthält, die beide einen Tag-Filter spezifizieren:

- Regel 1 spezifiziert einen Tag-basierten Filter (tag1/value1). Diese Regel weist Amazon S3 an, Objekte 365 Tage nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen.
- Regel 2 spezifiziert einen Tag-basierten Filter (tag2/value2). Diese Regel weist Amazon S3 an, Objekte 14 Tage nach der Erstellung ablaufen zu lassen.

Die S3-Lebenszyklus-Konfiguration wird im Folgenden angezeigt.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Tag>
        <Key>tag1</Key>
        <Value>value1</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>GLACIER</StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Tag>
        <Key>tag2</Key>
        <Value>value2</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>14</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

```
</Rule>  
</LifecycleConfiguration>
```

Wenn ein Objekt beide Tags hat, muss Amazon S3 entscheiden, welche Regel befolgt werden soll. In diesem Fall lässt Amazon S3 das Objekt 14 Tage nach der Erstellung ablaufen. Das Objekt wird entfernt und die Übergangsaktion wird daher nicht angewendet.

Important

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von [Löschmarkierungen](#).
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter [Beispiel 5: Überlappende Filter, widersprüchliche Lebenszyklus-Aktionen, und was Amazon S3 mit nichtversionierten Buckets macht](#).

Beispiel 6: Spezifikation einer Lebenszyklus-Konfigurationsregel für einen Bucket mit Versioning

Angenommen, Sie haben einen Versioning-fähigen Bucket, d. h. Sie haben für jedes Objekt eine aktuelle Version und keine oder mehr nicht aktuelle Versionen. (Weitere Informationen über das S3-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).) In diesem Beispiel möchten Sie den Verlauf eines Jahres verwalten und die nicht aktuellen Versionen löschen. S3 Lifecycle-Konfigurationen unterstützen die Aufbewahrung von 1 bis 100 Versionen eines beliebigen Objekts.

Um Speicherkosten zu sparen, sollten Sie nicht aktuelle Versionen 30 Tage, nachdem sie nicht mehr aktuell werden, auf S3 Glacier Flexible Retrieval verschieben (vorausgesetzt, diese nicht aktuellen Objekte sind kalte Daten, für die Sie keinen Echtzeitzugriff benötigen). Darüber hinaus gehen Sie davon aus, dass die Zugriffshäufigkeit der aktuellen Versionen 90 Tage nach der Erstellung

abnehmen wird. Sie könnten sich also dafür entscheiden, diese Objekte in die Speicherklasse S3 Standard-IA zu verschieben.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>90</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NewerNoncurrentVersions>5</NewerNoncurrentVersions>
      <NoncurrentDays>365</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Beispiel 7: Löschen abgelaufener Löschmarkierungen für Objekte

Ein Bucket mit Versioning enthält eine aktuelle Version und null oder mehr nicht aktuelle Versionen für jedes Objekt. Beachten Sie beim Löschen eines Objekts Folgendes:

- Wenn Sie keine Versions-ID in Ihrer Löschanfrage angeben, fügt Amazon S3 eine Löschmarkierung hinzu, statt das Objekt zu löschen. Das aktuelle Version wird nicht aktuell, und die Löschmarkierung wird zur aktuellen Version.
- Wenn Sie eine Versions-ID in Ihrer Löschanfrage angeben, löscht Amazon S3 die Objektversion permanent (es wird keine Löschmarkierung erstellt).
- Eine Löschmarkierung mit null nicht aktuellen Versionen wird als Löschmarkierung für das abgelaufene Objekt bezeichnet.

Dieses Beispiel zeigt ein Szenario, das Löschmarkierungen für abgelaufene Objekte in Ihrem Bucket erstellen kann, und demonstriert, wie Sie mit einer S3-Lebenszyklus-Konfiguration Amazon S3 anweisen können, die Löschmarkierungen für abgelaufene Objekte zu löschen.

Angenommen, Sie schreiben eine S3 Lifecycle-Konfiguration, die die `NoncurrentVersionExpiration` Aktion verwendet, um die nicht aktuellen Versionen 30 Tage, nachdem sie nicht mehr aktuell sind, zu entfernen und dabei maximal 10 nicht aktuelle Versionen beizubehalten, wie im folgenden Beispiel gezeigt.

```
<LifecycleConfiguration>
  <Rule>
    ...
    <NoncurrentVersionExpiration>
      <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Die Aktion `NoncurrentVersionExpiration` gilt nicht für die aktuellen Objektversionen. Sie entfernt nur nicht aktuelle Versionen.

Für aktuelle Objektversionen haben Sie die folgenden Optionen, ihre Lebensdauer zu verwalten, abhängig davon, ob die aktuellen Objektversionen einen definierten Lebenszyklus haben:

- Aktuelle Objektversionen folgen einem gut definierten Lebenszyklus.

In diesem Fall können Sie eine S3-Lebenszykluskonfiguration mit der Aktion `Expiration` verwenden, um Amazon S3 anzuweisen, die aktuellen Versionen zu entfernen, wie im folgenden Beispiel gezeigt.

```
<LifecycleConfiguration>
  <Rule>
    ...
    <Expiration>
      <Days>60</Days>
    </Expiration>
    <NoncurrentVersionExpiration>
      <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
```

```
</Rule>  
</LifecycleConfiguration>
```

In diesem Beispiel entfernt Amazon S3 aktuelle Versionen 60 Tage nach ihrer Erstellung, indem für jede der aktuellen Objektversionen eine Löschmarkierung hinzugefügt wird. Durch diesen Vorgang wird die aktuelle Version nicht mehr aktuell und die Löschmarkierung wird zur aktuellen Version. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Note

Sie können nicht sowohl ein `Days` als auch `ExpiredObjectDeleteMarker`-Tag für dieselbe Regel angeben. Wenn Sie das `Days`-Tag angeben, führt Amazon S3 automatisch eine `ExpiredObjectDeleteMarker`-Bereinigung durch, wenn die Löschmarkierungen alt genug sind, um die Alterskriterien zu erfüllen. Um Löschmarkierungen zu bereinigen, sobald sie die einzige Version werden, erstellen Sie eine separate Regel nur mit dem `ExpiredObjectDeleteMarker`-Tag.

Die `NoncurrentVersionExpiration`-Aktion in derselben S3-Lebenszyklus-Konfiguration entfernt nicht aktuelle Objekte 30 Tage, nachdem sie nicht aktuell wurden. Somit werden in diesem Beispiel 90 Tage nach der Objekterstellung alle Objektversionen dauerhaft entfernt. Obwohl während dieses Vorgangs Löschmarkierungen für abgelaufene Objekte erstellt werden, erkennt und entfernt Amazon S3 die Löschmarkierungen für abgelaufene Objekte für Sie.

- Aktuelle Objektversionen folgen keinem gut definierten Lebenszyklus.

In diesem Fall müssen Sie die Objekte möglicherweise manuell entfernen, wenn Sie sie nicht mehr brauchen, und eine Löschmarkierungen mit einer oder mehreren nicht aktuellen Versionen erstellen. Wenn Ihre S3-Lebenszyklus-Konfiguration mit der `NoncurrentVersionExpiration`-Aktion alle nicht aktuellen Versionen löscht, haben Sie jetzt Löschmarkierungen für abgelaufene Objekte.

Speziell für dieses Szenario stellt die S3-Lebenszykluskonfiguration die `Expiration`-Aktion bereit, mit der Sie die Löschmarkierungen für abgelaufene Objekte entfernen können.

```
<LifecycleConfiguration>  
  <Rule>  
    <ID>Rule 1</ID>
```



```
<Filter>
  <Prefix>logs/</Prefix>
</Filter>
<Status>Enabled</Status>
<Expiration>
  <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
</Expiration>
<NoncurrentVersionExpiration>
  <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
  <NoncurrentDays>30</NoncurrentDays>
</NoncurrentVersionExpiration>
</Rule>
</LifecycleConfiguration>
```

Setzen Sie das `ExpiredObjectDeleteMarker`-Element in der `Expiration`-Aktion auf `true`, um Amazon S3 anzuweisen, Löschmarkierungen für abgelaufene Objekte zu entfernen.

Note

Bei Verwendung der `ExpiredObjectDeleteMarker`-S3-Lebenszyklus-Aktion kann die Regel keinen Tag-basierten Filter angeben.

Beispiel 8: Lebenszyklus-Konfigurationsregel für das Abbrechen mehrteiliger Uploads

Sie können die mehrteiligen REST API Upload-Operationen von Amazon S3 verwenden, um große Objekte in Teilen hochzuladen. Weitere Informationen über mehrteilige Uploads finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Mithilfe einer S3-Lifecycle-Konfiguration können Sie Amazon S3 anweisen, unvollständige mehrteilige Uploads (identifiziert durch das in der Regel angegebene Schlüsselnamenpräfix) zu beenden, wenn sie nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wenn Amazon S3 einen mehrteiligen Upload abbricht, werden alle diesem mehrteiligen Upload zugeordneten Teile gelöscht. Dieser Prozess hilft, Ihre Speicherkosten zu kontrollieren, indem Sie sicherstellen, dass Sie keine unvollständigen mehrteiligen Uploads mit Teilen haben, die in Amazon S3 gespeichert sind.

Note

Bei Verwendung der `AbortIncompleteMultipartUpload`-S3-Lebenszyklus-Aktion kann die Regel keinen Tag-basierten Filter angeben.

Das folgende Beispiel zeigt eine S3-Lebenszyklus-Konfiguration, die eine Regel mit der Aktion `AbortIncompleteMultipartUpload` spezifiziert. Diese Aktion leitet Amazon S3 dazu, unvollständige mehrteilige Uploads sieben Tage nach der Initiierung abzubrechen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Filter>
      <Prefix>SomeKeyPrefix</Prefix>
    </Filter>
    <Status>rule-status</Status>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>7</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

Beispiel 9: Lebenszykluskonfiguration mit größenbasierten Regeln

Sie können Regeln erstellen, die Objekte nur basierend auf ihrer Größe übergehen. Sie können eine Mindestgröße (`ObjectSizeGreaterThan`) oder eine Maximalgröße (`ObjectSizeLessThan`) angeben, oder Sie können einen Bereich von Objektgrößen in Bytes angeben. Wenn Sie mehr als einen Filter verwenden, z. B. ein Präfix und eine Größenregel, müssen Sie die Filter in ein `<And>`-Element umfassen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition with a prefix and based on size</ID>
    <Filter>
      <And>
        <Prefix>tax</Prefix>
        <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
      </And>
    </Filter>
    <Status>Enabled</Status>
```

```

<Transition>
  <Days>365</Days>
  <StorageClass>GLACIER</StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>

```

Wenn Sie einen Bereich mit den `ObjectSizeGreaterThan`- und `ObjectSizeLessThan`-Elementen angeben, muss die maximale Objektgröße größer als die minimale Objektgröße sein. Wenn Sie mehr als einen Filter verwenden, müssen Sie die Filter in ein `<And>`-Element packen. Das folgende Beispiel zeigt, wie Objekte in einem Bereich zwischen 500 Byte und 64.000 Byte angegeben werden. Wenn Sie einen Bereich angeben, schließen die `ObjectSizeLessThan` Filter `ObjectSizeGreaterThan` und die angegebenen Werte aus. Weitere Informationen finden Sie unter [the section called "Filterelement"](#).

```

<LifecycleConfiguration>
  <Rule>
    ...
    <And>
      <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
      <ObjectSizeLessThan>64000</ObjectSizeLessThan>
    </And>
  </Rule>
</LifecycleConfiguration>

```

Sie können auch Regeln erstellen, um nicht aktuelle Objekte, die keine Daten enthalten, ausdrücklich ablaufen zu lassen, einschließlich nicht aktueller Löschmoderungsobjekte, die in einem Bucket mit aktivierter Versionsverwaltung erstellt wurden. Das folgende Beispiel verwendet die `NoncurrentVersionExpiration`-Aktion, um nicht aktuelle Versionen 30 Tage, nachdem sie nicht mehr aktuell sind, zu entfernen und höchstens 10 nicht aktuelle Versionen beizubehalten, wie im folgenden Beispiel gezeigt. Außerdem wird das `ObjectSizeLessThan`-Element verwendet, um nur Objekte ohne Daten zu filtern.

```

<LifecycleConfiguration>
  <Rule>
    <ID>Expire noncurrent with size less than 1 byte</ID>
    <Filter>
      <ObjectSizeLessThan>1</ObjectSizeLessThan>
    </Filter>
    <Status>Enabled</Status>
  </Rule>
</LifecycleConfiguration>

```

```
<NoncurrentVersionExpiration>
  <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
  <NoncurrentDays>30</NoncurrentDays>
</NoncurrentVersionExpiration>
</Rule>
</LifecycleConfiguration>
```

Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind

Amazon S3 wendet auf Ihre Lifecycle-Konfiguration ein Standardverhalten an, das verhindert, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden. Sie können den Übergang kleinerer Objekte zulassen, indem Sie der Konfiguration einen Filter für die Mindestgröße (`ObjectSizeGreaterThan`) oder die maximale Größe (`ObjectSizeLessThan`) hinzufügen, der eine kleinere Größe angibt. Das folgende Beispiel ermöglicht jedem Objekt, das kleiner als 128 KB ist, den Übergang zur S3 Glacier Instant Retrieval-Speicherklasse:

```
<LifecycleConfiguration>
  <Rule>
    <ID>Allow small object transitions</ID>
    <Filter>
      <ObjectSizeGreaterThan>1</ObjectSizeGreaterThan>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER_IR</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Note

Im September 2024 hat Amazon S3 das standardmäßige Übergangsverhalten für kleine Objekte wie folgt aktualisiert:

- Aktualisiertes Standard-Übergangsverhalten — Ab September 2024 verhindert das Standardverhalten, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden.

- Bisheriges Standard-Übergangsverhalten — Vor September 2024 erlaubte das Standardverhalten, dass Objekte, die kleiner als 128 KB waren, nur in die Speicherklassen S3 Glacier und S3 Glacier Deep Archive übertragen wurden.

Konfigurationen, die vor September 2024 erstellt wurden, behalten das vorherige Übergangsverhalten bei, sofern Sie sie nicht ändern. Das heißt, wenn Sie Regeln erstellen, bearbeiten oder löschen, ändert sich das standardmäßige Übergangsverhalten für Ihre Konfiguration in das aktualisierte Verhalten. Falls es Ihr Anwendungsfall erfordert, können Sie das standardmäßige Übergangsverhalten so ändern, dass Objekte, die kleiner als 128 KB sind, auf S3 Glacier und S3 Glacier Deep Archive umgestellt werden. Verwenden Sie dazu den optionalen `x-amz-transition-object-size-minimum-default` Header in einer [PutBucketLifecycleConfiguration](#)Anfrage.

Das folgende Beispiel zeigt, wie der `x-amz-transition-object-size-minimum-default` Header in einer [PutBucketLifecycleConfiguration](#)Anfrage verwendet wird, um das `varies_by_storage_class` standardmäßige Übergangsverhalten auf eine Lifecycle-Konfiguration anzuwenden. Dieses Verhalten ermöglicht Objekten, die kleiner als 128 KB sind, den Übergang zu den Speicherklassen S3 Glacier oder S3 Glacier Deep Archive. Standardmäßig verhindern alle anderen Speicherklassen Übergänge, die kleiner als 128 KB sind. Sie können weiterhin benutzerdefinierte Filter verwenden, um die minimale Übergangsgröße für jede Speicherklasse zu ändern. Benutzerdefinierte Filter haben immer Vorrang vor dem standardmäßigen Übergangsverhalten:

```
HTTP/1.1 200
x-amz-transition-object-size-minimum-default: varies_by_storage_class
<?xml version="1.0" encoding="UTF-8"?>
...
```

Amazon S3 Inventory

Sie können Amazon S3 Inventory verwenden, um Ihren Speicher zu verwalten. Sie können es beispielsweise für die Prüfung und Meldung des Replikations- und Verschlüsselungsstatus Ihrer Objekte für Unternehmens-, Compliance- und regulatorische Anforderungen verwenden. Sie können auch Geschäftsabläufe und Big-Data-Jobs vereinfachen und beschleunigen, indem Sie Amazon S3 Inventory verwenden, das eine geplante Alternative zu den synchronen List API Vorgängen von

Amazon S3 darstellt. Amazon S3 Inventory verwendet die List API Operationen nicht zur Prüfung Ihrer Objekte und hat keinen Einfluss auf die Anforderungsrate Ihres Buckets.

Amazon S3 Inventory bietet kommagetrennte Werte (,)CSV, [Apache optimierte Zeile spaltenförmig \(\)](#) oder [ORC Apache Parquet](#)Ausgabedateien, die Ihre Objekte und die entsprechenden Metadaten täglich oder wöchentlich für einen S3-Bucket oder Objekte mit einem gemeinsamen Präfix auflisten (d. h. Objekte, deren Namen mit einer gemeinsamen Zeichenfolge beginnen). Wenn Sie eine wöchentliche Bestandsaufnahme einrichten, wird jeden Sonntag (UTCZeitzone) nach dem ersten Bericht ein Bericht generiert. Informationen zu den Preisen für Amazon S3 Inventory finden Sie unter [Amazon-S3-Preise](#).

Sie können mehrere Bestandslisten für einen Bucket konfigurieren. Wenn Sie eine Bestandsliste konfigurieren, können Sie Folgendes angeben:

- welche Objektmetadaten in das Inventar aufgenommen werden sollen
- ob alle Objektversionen oder nur aktuelle Versionen aufgelistet werden sollen
- wo die Ausgabe der Bestandslistendatei gespeichert werden soll
- ob der Bestandsbericht täglich oder wöchentlich generiert werden soll
- ob die Bestandslistendatei verschlüsselt werden soll

Sie können Amazon S3 S3-Inventar mit SQL Standardabfragen abfragen, indem Sie [Amazon Athena](#), [Amazon Redshift Spectrum](#) und andere Tools verwenden, z. B. [Presto](#), [Apache Hive](#), und [Apache Spark](#). Weitere Informationen zur Verwendung von Athena zur Abfrage Ihrer Lagerbestandsdateien finden Sie unter [the section called "Bestandsabfrage mit Athena"](#).

Note

Es könnte bis zu 48 Stunden dauern, bis Amazon S3 den ersten Inventarbericht bereitstellt.

Quell- und Ziel-Buckets

Der Bucket, dessen Objekte die Bestandserfassung auflistet, wird als Quell-Bucket bezeichnet. Der Bucket, in dem die Datei mit der Bestandsliste gespeichert ist, wird als Ziel-Bucket bezeichnet.

Quell-Bucket

Der Bestand listet die Objekte auf, die im Quell-Bucket gespeichert sind. Sie können eine Bestandsliste für einen ganzen Bucket erhalten oder die Liste nach dem Präfix eines Objektschlüsselnamens filtern.

Der Quell-Bucket:

- Enthält die Objekte, die im Bestand aufgelistet sind.
- Enthält die Konfiguration für den Bestand.

Ziel-Bucket

Amazon S3 Inventory listet Dateien auf, die in den Ziel-Bucket geschrieben werden. Um alle Bestandslisten-Dateien an einem gemeinsamen Speicherort im Ziel-Bucket zu gruppieren, können Sie ein Zielpräfix in der Bestands-Konfiguration angeben.

Der Ziel-Bucket:

- Enthält die Dateilisten für den Bestand.
- Enthält die Manifestdateien, die alle Bestandslistendateien enthalten, die im Ziel-Bucket gespeichert sind. Weitere Informationen finden Sie unter [Bestandsmanifest](#).
- Erfordert eine Bucket-Richtlinie, um Amazon S3 die Berechtigung zu erteilen, die Eigentümerschaft an dem Bucket zu überprüfen, ebenso wie die Berechtigung, Dateien in den Bucket zu schreiben.
- Muss im selben sein AWS-Region wie der Quell-Bucket.
- Kann gleich dem Quell-Bucket sein.
- Kann einer anderen Person gehören AWS-Konto als das Konto, dem der Quell-Bucket gehört.

Amazon-S3-Inventory-Liste

Eine Datei mit einer Bestandsliste enthält eine Liste der Objekte im Quell-Bucket sowie die Metadaten für jedes Objekt. Eine Bestandslistendatei wird im Ziel-Bucket in einem der folgenden Formate gespeichert:

- Als komprimierte CSV Datei mit GZIP
- Als Apache optimierte row columnar (ORC) -Datei, komprimiert mit ZLIB
- Als Apache Parquet mit Snappy komprimierte Datei

 Note

Es kann nicht garantiert werden, dass Objekte in Berichten von Amazon S3 Inventory auf bestimmte Weise sortiert werden.

Eine Datei mit einer Bestandsliste enthält eine Liste der Objekte im Quell-Bucket sowie die Metadaten für jedes aufgelistete Objekt:

- Bucket name (Bucket-Name) – Der Name des Buckets, für den der Bestand gilt.
- Key name (Schlüsselname) – Name des Objektschlüssels (oder Schlüssel), der das Objekt in dem Bucket eindeutig identifiziert. Wenn Sie das CSV Dateiformat verwenden, ist der Schlüsselname URL -kodiert und muss dekodiert werden, bevor Sie ihn verwenden können.
- Version ID (Versions-ID) – ID der Objektversion. Wenn Sie das Versioning für einen Bucket aktivieren, weist Amazon S3 allen Objekten, die dem Bucket hinzugefügt werden, eine Versionsnummer zu. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#). (Dieses Feld ist nicht enthalten, wenn die Liste nur für die aktuelle Version der Objekte konfiguriert ist.)
- IsLatest— Legt fest, True ob es sich bei dem Objekt um die aktuelle Version des Objekts handelt. (Dieses Feld ist nicht enthalten, wenn die Liste nur für die aktuelle Version der Objekte konfiguriert ist.)
- Delete marker (Löschmarkierung) – Auf True gesetzt, wenn es sich bei dem Objekt um eine Löschmarkierung handelt. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#). (Dieses Feld wird dem Bericht automatisch hinzugefügt, wenn Sie den Bericht so konfiguriert haben, dass alle Versionen der Objekte aufgenommen werden.)
- Größe – Die Objektgröße in Byte, ohne die Größe von unvollständigen mehrteiligen Uploads, Objektmetadaten und Löschmarkierungen.
- Last modified date (Letztes Änderungsdatum) – Datum der Erstellung oder der letzten Änderung des Objekts, je nachdem, welches neuer ist.
- ETag— Das Entity-Tag (ETag) ist ein Hash des Objekts. Das ETag spiegelt nur Änderungen am Inhalt eines Objekts wider, nicht an seinen Metadaten. Das ETag kann eine MD5 Zusammenfassung der Objektdaten sein. Dies hängt davon ab, wie das Objekt erstellt und verschlüsselt wurde. Weitere Informationen finden Sie unter [Object](#) in der Amazon Simple Storage Service API Reference.

- **Storage class (Speicherklasse)** – Die für die Speicherung des Objekts verwendete Speicherklasse. Auf STANDARD, REDUCED_REDUNDANCY, STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, DEEP_ARCHIVE, OUTPOSTS, GLACIER_IR oder SNOW festgelegt. Weitere Informationen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

 Note

S3 Inventory unterstützt S3 Express One Zone nicht.

- **Multipart upload flag (Markierung für mehrteiligen Upload)** – Auf True gesetzt, wenn das Objekt als mehrteiliger Upload hochgeladen wurde. Weitere Informationen finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).
- **Replikationsstatus** – Auf PENDING, COMPLETED, FAILED, oder REPLICA gesetzt. Weitere Informationen finden Sie unter [Abrufen von Replikationsstatusinformationen](#).
- **Verschlüsselungsstatus** – Der serverseitige Verschlüsselungsstatus, abhängig von der Art des verwendeten Verschlüsselungsschlüssels— serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3), serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C). SSE Setzen Sie den Wert auf SSE-S3, SSE-KMS, DSSE-KMS, SSE-C oder NOT-SSE. Der Status NOT-SSE bedeutet, dass das Objekt nicht mit serverseitiger Verschlüsselung verschlüsselt ist. Weitere Informationen finden Sie unter [Datenschutz durch Verschlüsselung](#).
- **S3 Object Lock retain until date (Beibehaltungsfrist für S3-Objektsperre)** – Das Datum, bis zu dem das gesperrte Objekt nicht gelöscht werden kann. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).
- **S3 Object Lock retention mode (Aufbewahrungsmodus für S3-Objekt)** – Auf Governance oder Compliance gesetzt für Objekte, die gesperrt sind. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).
- **S3 Object Lock legal hold status (Rechtlicher Aufbewahrungsstatus der S3-Objektsperre)** – Auf On gesetzt, wenn für ein Objekt eine rechtliche Aufbewahrungsfrist gilt. Andernfalls lautet der Wert Off. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).
- **S3 Intelligent-Tiering access tier (S3-Intelligent-Tiering-Zugriffsebene)** – Zugriffsebene (häufig oder selten) des Objekts bei Speicherung in der Speicherklasse S3 Intelligent-Tiering. Setzen Sie den Wert auf FREQUENT, INFREQUENT, ARCHIVE_INSTANT_ACCESS, ARCHIVE oder DEEP_ARCHIVE. Weitere Informationen finden Sie unter [Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekanntem Zugriffsmustern](#).

- S3 Bucket Key status (Status des S3-Bucket-Schlüssels) – Auf ENABLED oder DISABLED gesetzt. Gibt an, ob das Objekt einen S3-Bucket-Key für - verwendet. SSE KMS Weitere Informationen finden Sie unter [Verwenden von Amazon-S3-Bucket-Schlüssel](#).
- Prüfsummen-Algorithmus – Gibt den Algorithmus an, mit dem die Prüfsumme für das Objekt erstellt wurde. Weitere Informationen finden Sie unter [Verwenden unterstützter Prüfsummenalgorithmen](#).
- Objektzugriffskontrollliste — Eine Zugriffskontrollliste (ACL) für jedes Objekt, die definiert, welche AWS-Konten oder Gruppen erhalten Zugriff auf dieses Objekt und die Art des Zugriffs, der gewährt wird. Das ACL Objektfeld ist im JSON Format definiert. Ein S3-Inventarbericht enthält ObjekteACLs, die Objekten in Ihrem Quell-Bucket zugeordnet ACLs sind, auch wenn sie für den Bucket deaktiviert sind. Weitere Informationen erhalten Sie unter [Arbeiten mit dem Feld „Objekt-ACL“](#) und [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Note

Das ACL Objektfeld ist im JSON Format definiert. In einem Inventarbericht wird der Wert für das ACL Objektfeld als Base64-kodierte Zeichenfolge angezeigt.

Nehmen wir beispielsweise an, Sie haben das folgende ACL Objektfeld im Format: JSON

```
{
  "version": "2022-11-10",
  "status": "AVAILABLE",
  "grants": [{
    "canonicalId": "example-canonical-user-ID",
    "type": "CanonicalUser",
    "permission": "READ"
  }]
}
```

Das ACL Objektfeld ist codiert und wird als folgende Base64-kodierte Zeichenfolge angezeigt:

```
eyJ2ZXJzaW9uIjoimjAyMi0xMS0xMCI5InN0YXR1cyI6IkkFWQUlMQUMRSIsImdyYW50cyI6W3siY2Fub25pY2Fs
```

Um den dekodierten Wert JSON für das ACL Objektfeld zu erhalten, können Sie dieses Feld in Amazon Athena abfragen. Abfragebeispiele finden Sie unter [Abfragen von Amazon S3 Inventory mit Amazon Athena](#).

- **Objekteigentümer** — Die kanonische Benutzer-ID des Besitzers des Objekts. Weitere Informationen [finden Sie unter Finden Sie die kanonische Benutzer-ID für Ihr AWS Konto](#) im AWS Referenzhandbuch zur Kontoverwaltung.

Note

Wenn ein Objekt basierend auf seiner Lebenszykluskonfiguration das Ende seiner Lebensdauer erreicht hat, stellt Amazon S3 das Objekt zum Entfernen in eine Warteschlange und entfernt es asynchron. Daher kann es eine Verzögerung zwischen dem Ablaufdatum und dem Datum geben, an dem Amazon S3 ein Objekt entfernt. Der Bestandsbericht enthält die Objekte, die abgelaufen sind, aber noch nicht entfernt wurden. Weitere Informationen über die Ablaufaktionen im S3-Lebenszyklus finden Sie unter [Auslaufende Objekte](#).

Im Folgenden finden Sie ein Beispiel für einen Inventarbericht mit zusätzlichen Metadatenfeldern, die aus vier Datensätzen bestehen.

```

amzn-s3-demo-bucket1  example-object-1  EXAMPLEDC81.XJCEN1F7LePaNIivs001  TRUE
  1500  2024-08-15T15:28:26.0004  EXAMPLE21e1518b92f3d92773570f600  STANDARD
  FALSE  COMPLETED  SSE-KMS  2025-01-25T15:28:26.000Z  COMPLIANCE  Off
ENABLED
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIzInN0YXR1cyI6IkFWQUMQUJMRSlzImdyYW50cyI6W3sicGVybWlzc2l2vbiI6Ik
EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7000
amzn-s3-demo-bucket1  example-object-2  EXAMPLEDC81.XJCEN1F7LePaNIivs002
TRUE  200  2024-08-21T15:28:26.000Z  EXAMPLE21e1518b92f3d92773570f601
INTELLIGENT_TIERING  FALSE  COMPLETED  SSE-KMS  2025-01-25T15:28:26.000Z
COMPLIANCE  Off  INFREQUENT  ENABLED  SHA-256
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIzInN0YXR1cyI6IkFWQUMQUJMRSlzImdyYW50cyI6W3sicGVybWlzc2l2vbiI6Ik
EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7001
amzn-s3-demo-bucket1  example-object-3  EXAMPLEDC81.XJCEN1F7LePaNIivs003  TRUE
  12500  2023-01-15T15:28:30.000Z  EXAMPLE21e1518b92f3d92773570f602  STANDARD
  FALSE  REPLICATA  SSE-KMS  2025-01-25T15:28:26.000Z  GOVERNANCE  On
ENABLED
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIzInN0YXR1cyI6IkFWQUMQUJMRSlzImdyYW50cyI6W3sicGVybWlzc2l2vbiI6Ik
EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7002
amzn-s3-demo-bucket1  example-object-4  EXAMPLEDC81.XJCEN1F7LePaNIivs004  TRUE
  100  2021-02-15T15:28:27.000Z  EXAMPLE21e1518b92f3d92773570f603  STANDARD
  FALSE  COMPLETED  SSE-KMS  2025-01-25T15:28:26.000Z  COMPLIANCE  Off
ENABLED

```

```
eyJ2ZXJzaW9uIjoieMjAyMi0xMS0xMCIzInN0YXR1cyI6IktFWQU1MQUJMRSIzImdyYW50cyI6W3sicGVybnw1zc2lvd2I6IktEXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7003
```

Wir empfehlen, eine Lebenszyklusrichtlinie einzurichten, die alte Bestandslisten löscht. Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Die `s3:PutInventoryConfiguration`-Berechtigung ermöglicht es einem Benutzer, beim Konfigurieren einer Bestandsliste sowohl alle Metadatenfelder auszuwählen, die zuvor für jedes Objekt aufgelistet wurden, als auch den Ziel-Bucket zum Speichern des Bestands anzugeben. Ein Benutzer mit Lesezugriff auf Objekte im Ziel-Bucket kann auf alle Objektmetadatenfelder zugreifen, die in der Bestandsliste verfügbar sind. Informationen zum Einschränken des Zugriffs auf Bestandsberichte finden Sie unter [Gewähren von Berechtigungen für S3 Inventory und S3 Analytics](#).

Bestandskonsistenz

Möglicherweise erscheinen nicht alle Ihre Objekte in jeder Bestandsliste. Die Bestandsliste bietet letztendliche Datenkonsistenz für PUT-Anforderungen (sowohl neuer als auch überschriebener Objekte) sowie für DELETE-Anforderungen. Jede Bestandsliste für einen Bucket ist ein Snapshot der Bucket-Elemente. Diese Listen sind letztendlich konsistent (das heißt, eine Liste enthält möglicherweise keine kürzlich hinzugefügten oder gelöschten Objekte).

Um den Status eines Objekts zu überprüfen, bevor Sie Maßnahmen für das Objekt ergreifen, empfehlen wir Ihnen, eine `HeadObject` REST API Anfrage zum Abrufen von Metadaten für das Objekt durchzuführen oder die Eigenschaften des Objekts in der Amazon S3 S3-Konsole zu überprüfen. Sie können die Objektmetadaten auch mit dem AWS CLI oder das AWS SDKS. Weitere Informationen finden Sie [HeadObject](#) in der Amazon Simple Storage Service API Reference.

Weitere Informationen zur Arbeit mit Amazon S3 Inventory finden Sie in den folgenden Themen.

Themen

- [Konfigurieren von Amazon S3 Inventory](#)
- [Lokalisieren Ihrer Bestandsliste](#)
- [Einrichten von Amazon-S3-Ereignis-Benachrichtigungen für den Bestandsabschluss](#)
- [Abfragen von Amazon S3 Inventory mit Amazon Athena](#)
- [Konvertieren leerer Versions-ID-Strings in Amazon-S3-Inventory-Berichten in Null-Zeichenfolgen](#)
- [Arbeiten mit dem Feld „Objekt-ACL“](#)

Konfigurieren von Amazon S3 Inventory

Amazon S3 Inventory stellt eine Flat-File-Liste Ihrer Objekte und Metadaten nach einem von Ihnen definierten Zeitplan bereit. Sie können S3 Inventory als geplante Alternative zum synchronen Amazon S3 List API S3-Betrieb verwenden. S3 Inventory bietet kommagetrennte Werte (), CSV [Apache optimierte Zeile spaltenförmig \(\)](#) oder ORC [Apache Parquet \(Parquet\)](#) Ausgabedateien, die Ihre Objekte und die entsprechenden Metadaten auflisten.

Sie können S3 Inventory so konfigurieren, dass täglich oder wöchentlich Bestandslisten für einen S3-Bucket oder für Objekte mit gemeinsamem Präfix (Objekte, deren Namen mit derselben Zeichenfolge beginnen) erstellt werden. Weitere Informationen finden Sie unter [Amazon S3 Inventory](#).

In diesem Abschnitt wird beschrieben, wie Sie einen Bestand, einschließlich Details über die Quell- und Ziel-Buckets des Bestands, einrichten.

Themen

- [Übersicht](#)
- [Erstellen einer Ziel-Bucket-Richtlinie](#)
- [Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung](#)
- [Konfigurieren des Bestands mit der S3-Konsole](#)
- [Verwenden von RESTAPI, um mit S3 Inventory zu arbeiten](#)

Übersicht

Amazon S3 Inventory hilft Ihnen, Ihren Speicher zu verwalten, indem nach einem definierten Zeitplan Listen der Objekte in einem S3-Bucket erstellt werden. Sie können mehrere Bestandslisten für einen Bucket konfigurieren. Die Inventarlisten werden veröffentlicht in CSV, ORC, oder Parquet Dateien in einem Ziel-Bucket.

Der einfachste Weg, ein Inventar einzurichten, ist die Verwendung der Amazon S3-Konsole, aber Sie können auch Amazon S3 REST API verwenden. AWS Command Line Interface (AWS CLI), oder AWS SDKs. Die Konsole führt den ersten Schritt des folgenden Verfahrens für Sie durch: Hinzufügen einer Bucket-Richtlinie zum Ziel-Bucket.

So richten Sie Amazon S3 Inventory für einen S3-Bucket ein

1. Fügen Sie eine Bucket-Richtlinie für den Ziel-Bucket hinzu.

Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket erstellen, die Amazon S3 die Erlaubnis erteilt, Objekte in den Bucket am definierten Speicherort zu schreiben. Eine Beispielrichtlinie finden Sie unter [Gewähren von Berechtigungen für S3 Inventory und S3 Analytics](#).

2. Konfigurieren Sie eine Bestandsliste, um die Objekte in einem Quell-Bucket aufzulisten und die Liste in einem Ziel-Bucket zu veröffentlichen.

Wenn Sie eine Bestandsliste für einen Quell-Bucket konfigurieren, geben Sie den Ziel-Bucket an, in dem die Liste gespeichert werden soll, und ob Sie möchten, dass die Liste täglich oder wöchentlich generiert werden soll. Sie können auch konfigurieren, ob alle Objektversionen oder nur aktuelle Versionen aufgeführt werden sollen und welche Objektmetadaten aufgenommen werden sollen.

Einige Objektmetadatenfelder in Konfigurationen für S3-Inventarberichte sind optional, d. h. sie sind standardmäßig verfügbar, können aber eingeschränkt werden, wenn Sie einem Benutzer die `s3:PutInventoryConfiguration` entsprechende Berechtigung erteilen. Mithilfe des `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssels können Sie steuern, ob Benutzer diese optionalen Metadatenfelder in ihre Berichte aufnehmen können.

Weitere Informationen zu den optionalen Metadatenfeldern, die in S3 Inventory verfügbar sind, finden Sie unter [OptionalFields](#) in der Amazon Simple Storage Service API Reference. Weitere Informationen zur Beschränkung des Zugriffs auf bestimmte optionale Metadatenfelder in einer Inventarkonfiguration finden Sie unter [Steuern Sie die Erstellung der Konfiguration des S3-Inventarberichts](#).

Sie können angeben, dass die Inventarlistendatei verschlüsselt werden soll, indem Sie serverseitige Verschlüsselung mit einem von Amazon S3 verwalteten Schlüssel (SSE-S3) oder einem AWS Key Management Service (AWS KMS) vom Kunden verwalteter Schlüssel (SSE-KMS).

Note

Das Tool Von AWS verwalteter Schlüssel (`aws/s3`) wird nicht unterstützt für SSE - KMS Verschlüsselung mit S3 Inventory.

Weitere Hinweise zu SSE -S3 und SSE - finden Sie KMS unter [Schützen von Daten mit serverseitiger Verschlüsselung](#). Wenn Sie beabsichtigen, die SSE KMS -Verschlüsselung zu verwenden, lesen Sie Schritt 3.

- Informationen zur Verwendung der Konsole zum Konfigurieren einer Bestandsliste finden Sie unter [Konfigurieren des Bestands mit der S3-Konsole](#).
 - Um Amazon S3 API zur Konfiguration einer Inventarliste zu verwenden, verwenden Sie den [PutBucketInventoryConfiguration](#) RESTAPI Operation oder das Äquivalent von AWS CLI or AWS SDKs.
3. Um die Inventarlistendatei mit SSE - zu verschlüsseln KMS, gewähren Sie Amazon S3 die Erlaubnis zur Verwendung von AWS KMS key.

Sie können die Verschlüsselung für die Inventarlistendatei mithilfe der Amazon S3-Konsole, Amazon S3 RESTAPI, konfigurieren. AWS CLI, oder AWS SDKs. Unabhängig davon, welche Möglichkeit Sie wählen: Sie müssen Amazon S3 die Berechtigung gewähren, den vom Kunden verwalteten Schlüssel zum Verschlüsseln der Bestandsdatei zu verwenden. Sie erteilen Amazon S3 die Berechtigung, indem Sie die Schlüsselrichtlinie für den kundenverwalteten Schlüssel ändern, den Sie zur Verschlüsselung der Bestandsdatei verwenden möchten. Weitere Informationen finden Sie unter [Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung](#).

Der Ziel-Bucket, in dem die Inventarlistendatei gespeichert ist, kann einer anderen Person gehören AWS-Konto als das Konto, dem der Quell-Bucket gehört. Wenn Sie SSE — KMS Verschlüsselung für den kontoübergreifenden Betrieb von Amazon S3 S3-Inventar verwenden, empfehlen wir Ihnen, ARN bei der Konfiguration des S3-Inventars einen vollqualifizierten KMS Schlüssel zu verwenden. Weitere Informationen finden Sie unter [Verwendung von SSE — KMS Verschlüsselung für kontoübergreifende Operationen](#) und [ServerSideEncryptionByDefault](#) in der Amazon Simple Storage Service API Reference.

Erstellen einer Ziel-Bucket-Richtlinie

Wenn Sie Ihre Bestandskonfiguration über die S3-Konsole erstellen, erstellt Amazon S3 automatisch eine Bucket-Richtlinie für den Ziel-Bucket, die ihm Amazon-S3-Schreibberechtigung gewährt. Wenn Sie Ihre Inventarkonfiguration jedoch über die AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI, Sie müssen manuell eine Bucket-Richtlinie für den Ziel-Bucket hinzufügen. Die Richtlinie für S3-Inventar-Ziel-Buckets ermöglicht es Amazon S3, Daten für die Inventarberichte in den Bucket zu schreiben.

Im Folgenden finden Sie ein Beispiel für eine Bucket-Richtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "InventoryExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
      ],
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
        },
        "StringEquals": {
          "aws:SourceAccount": "source-account-id",
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Gewähren von Berechtigungen für S3 Inventory und S3 Analytics](#).

Wenn beim Erstellen der Bucket-Richtlinie ein Problem auftritt, erhalten Sie Anweisungen zur Fehlerbehebung. Wenn Sie beispielsweise einen Ziel-Bucket in einem anderen auswählen AWS-Konto und Sie haben keine Lese- und Schreibberechtigungen für die Bucket-Richtlinie, wird eine Fehlermeldung angezeigt.

In diesem Fall muss der Besitzer des Ziel-Buckets die Bucket-Richtlinie zum Ziel-Bucket hinzufügen. Wird die Richtlinie dem Ziel-Bucket nicht hinzugefügt, erhalten Sie keinen Bestandsbericht, da Amazon S3 nicht über die Berechtigung verfügt, in den Ziel-Bucket zu schreiben. Wenn der Quell-Bucket nicht dem Konto des aktuellen Benutzers gehört, muss die richtige Konto-ID des Quell-Bucket-Eigentümers in der Richtlinie ersetzt werden.

Note

Stellen Sie sicher, dass der Ziel-Bucket-Richtlinie keine Deny-Anweisungen hinzugefügt wurden, die die Übermittlung von Inventarberichten in diesen Bucket verhindern würden. Weitere Informationen finden Sie unter [Warum kann ich keinen Amazon S3 S3-Inventarbericht erstellen?](#) .

Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung

Um Amazon S3 die Erlaubnis zur Nutzung Ihres zu erteilen AWS Key Management Service (AWS KMS) Vom Kunden verwalteter Schlüssel für serverseitige Verschlüsselung müssen Sie eine Schlüsselrichtlinie verwenden. Gehen Sie wie folgt vor, um Ihre Schlüsselrichtlinie zu aktualisieren, damit Sie einen vom Kunden verwalteten Schlüssel verwenden können.

Um Amazon S3 Berechtigungen zur Verschlüsselung mithilfe Ihres vom Kunden verwalteten Schlüssels zu gewähren

1. Verwendung der AWS-Konto dem der vom Kunden verwaltete Schlüssel gehört, melden Sie sich bei AWS Management Console.
2. Öffnen Sie AWS KMS Konsole unter <https://console.aws.amazon.com/kms>.
3. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
4. Klicken Sie im linken Navigationsbereich auf Customer managed keys (Vom Kunden verwaltete Schlüssel).
5. Wählen Sie unter Kundenverwaltete Schlüssel den vom Kunden verwalteten Schlüssel aus, den Sie zum Verschlüsseln der Bestandsdateien verwenden möchten.
6. Wählen Sie im Abschnitt Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
7. Um die Schlüsselrichtlinie zu aktualisieren, wählen Sie Edit (Bearbeiten).
8. Fügen Sie auf der Seite Schlüsselrichtlinie bearbeiten die folgenden Zeilen zu der vorhandenen Schlüsselrichtlinie hinzu. Geben Sie für *source-account-id* und *amzn-s3-demo-source-bucket* die entsprechenden Werte für Ihren Anwendungsfall an.

```
{
```

```
"Sid": "Allow Amazon S3 use of the customer managed key",
"Effect": "Allow",
"Principal": {
  "Service": "s3.amazonaws.com"
},
"Action": [
  "kms:GenerateDataKey"
],
"Resource": "*",
"Condition":{
  "StringEquals":{
    "aws:SourceAccount": "source-account-id"
  },
  "ArnLike":{
    "aws:SourceARN": "arn:aws:s3:::amzn-s3-demo-source-bucket"
  }
}
}
```

9. Wählen Sie Änderungen speichern.

Weitere Informationen zur Erstellung von kundenverwalteten Schlüsseln und zur Verwendung von Schlüsselrichtlinien finden Sie unter den folgenden Links in der AWS Key Management Service Leitfaden für Entwickler:

- [Schlüssel verwalten](#)
- [Wichtige Richtlinien in AWS KMS](#)

Note

Stellen Sie sicher, dass der Richtlinie für das Ziel-Bucket keine Deny-Anweisungen hinzugefügt wurden, die die Übermittlung von Inventarberichten an diesen Bucket verhindern würden. Weitere Informationen finden Sie unter [Warum kann ich keinen Amazon S3 S3-Inventarbericht erstellen?](#) .

Konfigurieren des Bestands mit der S3-Konsole

Verwenden Sie diese Anweisungen, um den Bestand mit der S3-Konsole zu konfigurieren.

 Note

Es könnte bis zu 48 Stunden dauern, bis Amazon S3 den ersten Inventarbericht bereitstellt.


1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus. Wählen Sie in der Liste Bucket den Namen des Buckets aus, für den Sie Amazon S3 Inventory konfigurieren wollen.
3. Wählen Sie den Tab Management.
4. Wählen Sie unter Inventory configurations (Bestands-Konfigurationen) die Option Create inventory configuration (Bestands-Konfiguration erstellen).
5. Geben Sie unter Name der Bestandskonfiguration einen Namen ein.
6. Gehen Sie für Umfang für den Bestand wie folgt vor:
 - Geben Sie ein optionales Präfix ein.
 - Wählen Sie aus, welche Objektversionen eingeschlossen werden sollen: Nur aktuelle Versionen oder Alle Versionen einschließen.
7. Wählen Sie unter Berichtsdetails den Standort des AWS-Konto in dem Sie die Berichte speichern möchten: In diesem Konto oder in einem anderen Konto.
8. Wählen Sie unter Destination (Ziel) den Ziel-Bucket aus, in dem die Bestandsberichte gespeichert werden sollen.

Der Ziel-Bucket muss sich im selben Bucket befinden AWS-Region wie der Bucket, für den Sie das Inventar einrichten. Der Ziel-Bucket kann sich in einem anderen befinden AWS-Konto. Bei der Angabe des Ziel-Buckets können Sie auch ein optionales Präfix angeben, um Ihre Inventarberichte zu gruppieren.

Unter dem Bucket-Feld Destination (Ziel) sehen Sie die Anweisung Ziel-Bucket-Berechtigung, die der Ziel-Bucket-Richtlinie hinzugefügt wird, damit Amazon S3 Daten in diesen Bucket platzieren kann. Weitere Informationen finden Sie unter [Erstellen einer Ziel-Bucket-Richtlinie](#).

9. Wählen Sie unter Häufigkeit aus, wie oft der Bericht erstellt wird: Täglich oder Wöchentlich.
10. Wählen Sie als Ausgabeformat eines der folgenden Formate für den Bericht aus:

- CSV— Wenn Sie diesen Inventarbericht mit S3 Batch Operations verwenden möchten oder wenn Sie diesen Bericht in einem anderen Tool wie Microsoft Excel analysieren möchten, wählen Sie CSV.
 - Apache ORC
 - Apache Parquet
11. Wählen Sie unter Status die Option Enable (Aktivieren) oder Disable (Deaktivieren) aus.
 12. Führen Sie unter Verschlüsselung des Bestandsberichts die folgenden Schritte aus, um die serverseitige Verschlüsselung zu konfigurieren:
 - a. Wählen Sie unter Serverseitige Verschlüsselung entweder Keinen Verschlüsselungsschlüssel angeben oder Einen Verschlüsselungsschlüssel zum Verschlüsseln von Daten angeben aus.
 - Um die Bucket-Einstellungen für die serverseitige Standardverschlüsselung von Objekten beizubehalten, wenn sie in Amazon S3 gespeichert werden, wählen Sie Keinen Verschlüsselungsschlüssel angeben aus. Solange S3-Bucket-Schlüssel für den Ziel-Bucket aktiviert sind, wendet der Kopiervorgang S3-Bucket-Schlüssel auf den Ziel-Bucket an.


 Note

Wenn die Bucket-Richtlinie für das angegebene Ziel vorschreibt, dass Objekte verschlüsselt werden müssen, bevor sie in Amazon S3 gespeichert werden, müssen Sie „Verschlüsselungsschlüssel angeben“ wählen. Andernfalls schlägt das Kopieren von Objekten in das Ziel fehl.

- Um Objekte zu verschlüsseln, bevor sie in Amazon S3 gespeichert werden, wählen Sie Verschlüsselungsschlüssel angeben aus.
- b. Wenn Sie unter Verschlüsselungstyp die Option Verschlüsselungsschlüssel angeben ausgewählt haben, müssen Sie entweder den von Amazon S3 verwalteten Schlüssel (SSE-S3) oder AWS Key Management Service Schlüssel (SSE-KMS).


SSE-S3 verwendet eine der stärksten Blockchiffren — den 256-Bit-Advanced Encryption Standard (AES-256), um jedes Objekt zu verschlüsseln. SSE- bietet Ihnen mehr Kontrolle KMS über Ihren Schlüssel. Weitere Hinweise zu SSE -S3 finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#). Weitere

Hinweise zu SSE - finden Sie KMS unter [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

 Note


Um die Inventarlistendatei mit SSE - zu verschlüsseln KMS, müssen Sie Amazon S3 die Erlaubnis erteilen, den vom Kunden verwalteten Schlüssel zu verwenden. Eine Anleitung finden Sie unter [Amazon S3 Permission to Encrypt Using Your KMS Keys gewähren](#).

- c. Wenn Sie wählen AWS Key Management Service Taste (SSE-KMS), unter AWS KMS key, können Sie Ihre angeben AWS KMS geben Sie eine der folgenden Optionen ein.

 Note

Wenn der Ziel-Bucket, in dem die Inventarlistendatei gespeichert ist, einem anderen gehört AWS-Konto, stellen Sie sicher, dass Sie einen vollqualifizierten KMS Schlüssel verwenden ARN, um Ihren KMS Schlüssel anzugeben.

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS Schlüssel und wählen Sie einen symmetrischen KMS Verschlüsselungsschlüssel aus der Liste der verfügbaren Schlüssel aus. Stellen Sie sicher, dass sich der KMS Schlüssel in derselben Region wie Ihr Bucket befindet.

 Note

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in der Liste angezeigt. Allerdings ist der Von AWS verwalteter Schlüssel (aws/s3) wird nicht unterstützt für SSE - KMS Verschlüsselung mit S3 Inventory.

- Um den KMS Schlüssel einzugeben ARN, wählen Sie Enter AWS KMS Schlüssel ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen, klicken Sie auf AWS KMS Konsole, wählen Sie Create a KMS Key aus.

13. Wählen Sie für Zusätzliche Metadatenfelder eine oder mehrere der folgenden Optionen für das Hinzufügen zum Bestandsbericht aus:
- Größe – Die Objektgröße in Byte, ohne die Größe von unvollständigen mehrteiligen Uploads, Objektmetadaten und Löschmarkierungen.
 - Last modified date (Letztes Änderungsdatum) – Datum der Erstellung oder der letzten Änderung des Objekts, je nachdem, welches neuer ist.
 - Multipart upload (Mehrteiliger Upload) – Gibt an, dass das Objekt als mehrteiliger Upload hochgeladen wurde. Weitere Informationen finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).
 - Replication status (Replikationsstatus) – Der Replikationsstatus des Objekts. Weitere Informationen finden Sie unter [Abrufen von Replikationsstatusinformationen](#).
 - Encryption status (Verschlüsselungsstatus) – Der serverseitige Verschlüsselungstyp, der für die Verschlüsselung des Objekts verwendet wird. Weitere Informationen finden Sie unter [Schützen von Daten mit serverseitiger Verschlüsselung](#).
 - Bucket-Key-Status — Gibt an, ob ein Schlüssel auf Bucket-Ebene generiert wurde von AWS KMS gilt für das Objekt. Weitere Informationen finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#).
 - Objektzugriffskontrollliste — Eine Zugriffskontrollliste (ACL) für jedes Objekt, die definiert, welche AWS-Konten oder Gruppen erhalten Zugriff auf dieses Objekt und die Art des Zugriffs, der gewährt wird. Weitere Hinweise zu diesem Feld finden Sie unter [Arbeiten mit dem Feld „Objekt-ACL“](#). Weitere Informationen zu finden ACLs Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).
 - Object owner (Besitzer des Objekts) – Der Besitzer des Objekts.
 - Storage class (Speicherklasse) – Die für die Speicherung des Objekts verwendete Speicherklasse.
 - Intelligent-Tiering: Access tier (Zugriffsebene) – Zugriffsebene (häufig oder selten) des Objekts, wenn es im S3-Intelligent-Tiering-Speicher gespeichert wurde. Weitere Informationen finden Sie unter [Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekanntem Zugriffsmustern](#).
 - ETag— Das Entity-Tag (ETag) ist ein Hash des Objekts. Das ETag spiegelt nur Änderungen am Inhalt eines Objekts wider, nicht an seinen Metadaten. Dies ETag kann eine MD5 Zusammenfassung der Objektdaten sein oder auch nicht. Dies hängt davon ab, wie das Objekt erstellt und verschlüsselt wurde. Weitere Informationen finden Sie unter [Object](#) in der Amazon Simple Storage Service API Reference.

- Prüfsummen-Algorithmus – Gibt den Algorithmus an, mit dem die Prüfsumme für das Objekt erstellt wurde. Weitere Informationen finden Sie unter [Verwenden unterstützter Prüfsummenalgorithmen](#).
- All Object lock configurations (Alle Objektsperre-Konfigurationen) – Der Objektsperrenstatus des Objekts, einschließlich der folgenden Einstellungen:
 - Object Lock: Retention mode (Aufbewahrungsmodus der Objektsperre) – Die auf das Objekt angewendete Schutzebene, entweder Governance oder Compliance.
 - Object Lock: Retain until date (Beibehaltungsfrist für Objektsperre) – Das Datum, bis zu dem das gesperrte Objekt nicht gelöscht werden kann.
 - Object Lock: Legal hold status (Status der gesetzlichen Sperrfrist des Objekts) – Der Status der gesetzlichen Sperrfrist des gesperrten Objekts.

Weitere Informationen zur S3-Objektsperre finden Sie unter [So funktioniert die S3-Objektsperre](#).

Weitere Informationen zum Inhalt eines Bestandsberichts finden Sie unter [Amazon-S3-Inventory-Liste](#).

Weitere Informationen zur Beschränkung des Zugriffs auf bestimmte optionale Metadatenfelder in einer Inventarkonfiguration finden Sie unter [Steuern Sie die Erstellung der Konfiguration des S3-Inventarberichts](#).

14. Wählen Sie Create (Erstellen) aus.

Verwenden von RESTAPI, um mit S3 Inventory zu arbeiten

Die folgenden REST Operationen können Sie verwenden, um mit Amazon S3 Inventory zu arbeiten.

- [DeleteBucketInventoryConfiguration](#)
- [GetBucketInventoryConfiguration](#)
- [ListBucketInventoryConfigurations](#)
- [PutBucketInventoryConfiguration](#)

Lokalisieren Ihrer Bestandsliste

Wenn eine Bestandsliste veröffentlicht wird, werden die Manifestdateien am folgenden Standort im Ziel-Bucket veröffentlicht.

```
destination-prefix/source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.json  
destination-prefix/source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.checksum  
destination-prefix/source-bucket/config-ID/hive/dt=YYYY-MM-DD-HH-MM/symlink.txt
```

- *destination-prefix* ist das Präfix des Objektschlüsselnamens, das optional in der Bestandskonfiguration festgelegt wird. Sie können dieses Präfix verwenden, um alle Bestandslisten-Dateien an einem gemeinsamen Standort im Ziel-Bucket zu gruppieren.
- *source-bucket* ist der Quell-Bucket, für den die Bestandsliste erstellt wird. Der Quell-Bucket-Name wird hinzugefügt, um Kollisionen zu vermeiden, wenn mehrere Bestandsberichte von mehreren Quell-Buckets an denselben Ziel-Bucket gesendet werden.
- *config-ID* wird hinzugefügt, um Kollisionen zu vermeiden, wenn mehrere Bestandsberichte vom selben Quell-Bucket an denselben Ziel-Bucket gesendet werden. Die *config-ID* kommt aus der Bestandsbericht-Konfiguration und ist der Name des Berichts, der bei der Einrichtung festgelegt wurde.
- *YYYY-MM-DDTHH-MMZ* ist der Zeitstempel, der sich aus der Startzeit und dem Datum zusammensetzt, an dem die Bestandsberichtserstellung mit dem Scannen des Buckets beginnt, z. B. 2016-11-06T21-32Z.
- *manifest.json* ist die Manifestdatei.
- *manifest.checksum* ist der MD5 Hash des Inhalts der *manifest.json* Datei.
- *symlink.txt* ist der Apache Hive-kompatible Manifestdatei.

Die Bestandslisten werden täglich oder wöchentlich am folgenden Standort im Ziel-Bucket veröffentlicht.

```
destination-prefix/source-bucket/config-ID/data/example-file-name.csv.gz  
...  
destination-prefix/source-bucket/config-ID/data/example-file-name-1.csv.gz
```

- *destination-prefix* ist das Präfix des Objektschlüsselnamens, das optional in der Bestandskonfiguration festgelegt wird. Es kann verwendet werden, um alle Bestandslisten-Dateien an einem gemeinsamen Standort innerhalb des Ziel-Buckets zu gruppieren.

- *source-bucket* ist der Quell-Bucket, für den die Bestandsliste erstellt wird. Der Quell-Bucket-Name wird hinzugefügt, um Kollisionen zu vermeiden, wenn mehrere Bestandsberichte von mehreren Quell-Buckets an denselben Ziel-Bucket gesendet werden.
- *example-file-name.csv.gz* ist eine der CSV Inventardateien. ORC Inventarnamen enden mit der Dateinamenerweiterung `.orc` und Parquet Inventarnamen enden mit der Dateinamenerweiterung `.parquet`.

Bestandsmanifest

Die Manifest-Dateien `manifest.json` und `symlink.txt` beschreiben, wo sich die Bestandsdateien befinden. Wenn eine neue Bestandsliste geliefert wird, wird sie durch eine neue Reihe von Manifestdateien begleitet. Diese Dateien könnten sich gegenseitig überschreiben. In versionierungsfähigen Buckets erstellt Amazon S3 neue Versionen der Manifestdateien.

Jedes in der `manifest.json`-Datei enthaltene Manifest bietet Metadaten und andere grundlegende Informationen zu einem Bestand. Diese Informationen beinhalten Folgendes:

- Name des Quell-Buckets
- Name des Ziel-Buckets
- Bestandsversion
- Erstellungszeitstempel im Epochen-Datumsformat, der aus der Startzeit und dem Datum besteht, an dem die Bestandberichtserstellung beginnt, den Bucket zu scannen
- Format und Schema der Bestandsdateien
- Liste der Bestandsdateien, die im Ziel-Bucket enthalten sind

Immer wenn eine `manifest.json` Datei geschrieben wird, wird sie von einer `manifest.checksum` Datei begleitet, die den MD5 Hash des `manifest.json` Dateiinhalts darstellt.

Example Bestandsmanifest in einer **`manifest.json`**-Datei

Die folgenden Beispiele zeigen ein Inventarmanifest in einer `manifest.json` Datei für CSVORC, und Parquet-formatierte Inventare.

CSV

Im Folgenden finden Sie ein Beispiel für ein Manifest in einer `manifest.json` Datei für ein CSV Inventar im -Format.

```
{
  "sourceBucket": "example-source-bucket",
  "destinationBucket": "arn:aws:s3:::example-inventory-destination-bucket",
  "version": "2016-11-30",
  "creationTimestamp" : "1514944800000",
  "fileFormat": "CSV",
  "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
Size, LastModifiedDate, ETag, StorageClass, IsMultipartUploaded,
ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode,
ObjectLockLegalHoldStatus, IntelligentTieringAccessTier, BucketKeyStatus,
ChecksumAlgorithm, ObjectAccessControlList, ObjectOwner",
  "files": [
    {
      "key": "Inventory/example-source-bucket/2016-11-06T21-32Z/
files/939c6d46-85a9-4ba8-87bd-9db705a579ce.csv.gz",
      "size": 2147483647,
      "MD5checksum": "f11166069f1990abeb9c97ace9cdfabc"
    }
  ]
}
```

ORC

Das Folgende ist ein Beispiel für ein Manifest in einer manifest.json Datei für ein Inventar im ORC -Format.

```
{
  "sourceBucket": "example-source-bucket",
  "destinationBucket": "arn:aws:s3:::example-destination-bucket",
  "version": "2016-11-30",
  "creationTimestamp" : "1514944800000",
  "fileFormat": "ORC",
  "fileSchema":
"struct<bucket:string,key:string,version_id:string,is_latest:boolean,is_delete_marker:boolean>"
  "files": [
    {
      "key": "inventory/example-source-bucket/data/
d794c570-95bb-4271-9128-26023c8b4900.orc",
      "size": 56291,
      "MD5checksum": "5925f4e78e1695c2d020b9f6eexample"
    }
  ]
}
```

```
}
```

Parquet

Das Folgende ist ein Beispiel für ein Manifest in einer `manifest.json` Datei für ein Parquet-formatiertes Inventar.

```
{
  "sourceBucket": "example-source-bucket",
  "destinationBucket": "arn:aws:s3:::example-destination-bucket",
  "version": "2016-11-30",
  "creationTimestamp": "1514944800000",
  "fileFormat": "Parquet",
  "fileSchema": "message s3.inventory { required binary bucket (UTF8);
required binary key (UTF8); optional binary version_id (UTF8); optional boolean
is_latest; optional boolean is_delete_marker; optional int64 size; optional
int64 last_modified_date (TIMESTAMP_MILLIS); optional binary e_tag (UTF8);
optional binary storage_class (UTF8); optional boolean is_multipart_uploaded;
optional binary replication_status (UTF8); optional binary encryption_status
(UTF8); optional int64 object_lock_retain_until_date (TIMESTAMP_MILLIS); optional
binary object_lock_mode (UTF8); optional binary object_lock_legal_hold_status
(UTF8); optional binary intelligent_tiering_access_tier (UTF8); optional binary
bucket_key_status (UTF8); optional binary checksum_algorithm (UTF8); optional
binary object_access_control_list (UTF8); optional binary object_owner (UTF8);}",
  "files": [
    {
      "key": "inventory/example-source-bucket/data/
d754c470-85bb-4255-9218-47023c8b4910.parquet",
      "size": 56291,
      "MD5checksum": "5825f2e18e1695c2d030b9f6eexample"
    }
  ]
}
```

Die `symlink.txt` Datei ist eine Apache Hive-kompatible Manifestdatei, die Folgendes ermöglicht: Hive um Inventardateien und die zugehörigen Datendateien automatisch zu erkennen. Das Tool Hive-kompatible Manifest funktioniert mit dem Hive-kompatiblen Dienst Athena und Amazon Redshift Spectrum. Es funktioniert auch mit Hive-kompatiblen Anwendungen, einschließlich [Presto](#), [Apache Hive](#), [Apache Spark](#) und viele andere.

⚠ Important

Die `symlink.txt` Apache Hive-kompatible Manifestdatei funktioniert derzeit nicht mit AWS Glue.

Lesen der `symlink.txt` Datei mit [Apache Hive](#) und [Apache Spark](#) wird nicht unterstützt für ORC und Parquet-formatierte Inventardateien.

Einrichten von Amazon-S3-Ereignis-Benachrichtigungen für den Bestandsabschluss

Sie können eine Amazon S3-Ereignis-Benachrichtigung einrichten, um informiert zu werden, wenn die Prüfsummendatei für das Manifest erstellt wird, woran zu erkennen ist, dass dem Ziel-Bucket eine Bestandsliste hinzugefügt wurde. Das Manifest ist eine up-to-date Liste aller Inventarlisten am Zielort.

Amazon S3 kann Ereignisse in einem Amazon Simple Notification Service (AmazonSNS) -Thema, einer Amazon Simple Queue Service (AmazonSQS) -Warteschlange oder einem AWS Lambda Funktion. Weitere Informationen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Die folgende Benachrichtigungskonfiguration definiert, dass alle dem Ziel-Bucket neu hinzugefügten `manifest.checksum` Dateien von der verarbeitet werden AWS Lambda `cloud-function-list-write`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>destination-prefix/source-bucket</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>checksum</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <CloudFunction>arn:aws:lambda:us-west-2:222233334444:cloud-function-list-write</CloudFunction>
  </QueueConfiguration>
</NotificationConfiguration>
```

```
<Event>s3:ObjectCreated:*</Event>
</QueueConfiguration>
</NotificationConfiguration>
```

Weitere Informationen finden Sie unter [Verwenden AWS Lambda mit Amazon S3](#) in der AWS Lambda Leitfaden für Entwickler.

Abfragen von Amazon S3 Inventory mit Amazon Athena

Sie können Amazon S3 S3-Inventardateien mit SQL Standardabfragen abfragen, indem Sie Amazon Athena in allen Regionen verwenden, in denen Athena verfügbar ist. [Informationen zur AWS-Region Verfügbarkeit finden Sie in der AWS-Region Tabelle.](#)

Athena kann Amazon S3 S3-Inventardateien im [Apacheoptimierten Format für Zeilen, Apache ParquetSpalten \(ORC\)](#) oder kommagetrennte Werte () abfragen. CSV Wenn Sie Athena verwenden, um Inventardateien abzufragen, empfehlen wir die Verwendung von -formatierten oder ORC Parquet -formatierten Inventardateien. Die Parquet Formate ORC und sorgen für eine schnellere Abfrageleistung und geringere Abfragekosten. ORC und Parquet sind selbsterklärende, typspezifische spaltenorientierte Dateiformate, für die sie konzipiert wurden. [Apache Hadoop](#) Das spaltenbasierte Format lässt den Leser nur die Spalten lesen, entpacken und verarbeiten, die für die aktuelle Abfrage benötigt werden. Die Parquet Formate ORC und für Amazon S3 Inventory sind in allen verfügbar AWS-Regionen.

So fragen Sie Amazon-S3-Inventory-Dateien mit Athena ab

1. Erstellen Sie eine Athena-Tabelle. Informationen zum Erstellen einer Tabelle finden Sie unter [Erstellen von Tabellen in Amazon Athena](#) im Amazon Athena-Benutzerhandbuch.
2. Erstellen Sie Ihre Abfrage anhand einer der folgenden Beispielabfragevorlagen, je nachdem, ob Sie einen ORC Lagerbestandsbericht im -Format, im Parquet-Format oder im -Format abfragen. CSV
 - Wenn Sie Athena verwenden, um einen Inventarbericht im ORC -Format abzufragen, verwenden Sie die folgende Beispielabfrage als Vorlage.

Die folgende Beispielabfrage enthält alle optionalen Felder in einem Inventarbericht im ORC -Format.

Gehen Sie wie folgt vor, um diese Beispielabfrage zu verwenden:

- Ersetzen Sie *your_table_name* durch den Namen der Athena-Tabelle, die Sie erstellt haben.
- Entfernen Sie alle optionalen Felder, die Sie für Ihren Bestand nicht ausgewählt haben, sodass die Abfrage den für Ihren Bestand gewählten Feldern entspricht.
- Ersetzen Sie den folgenden Bucket-Namen und den Bestandsort (die Konfigurations-ID) entsprechend Ihrer Konfiguration.

`s3://amzn-s3-demo-bucket/config-ID/hive/`

- Ersetzen Sie das Datum `2022-01-01-00-00` unter `projection.dt.range` mit dem ersten Tag des Zeitbereichs, innerhalb dessen Sie die Daten in Athena partitionieren. Weitere Informationen finden Sie unter [Partitionieren von Daten in Athena](#).

```
CREATE EXTERNAL TABLE your_table_name(
    bucket string,
    key string,
    version_id string,
    is_latest boolean,
    is_delete_marker boolean,
    size bigint,
    last_modified_date timestamp,
    e_tag string,
    storage_class string,
    is_multipart_uploaded boolean,
    replication_status string,
    encryption_status string,
    object_lock_retain_until_date bigint,
    object_lock_mode string,
    object_lock_legal_hold_status string,
    intelligent_tiering_access_tier string,
    bucket_key_status string,
    checksum_algorithm string,
    object_access_control_list string,
    object_owner string
) PARTITIONED BY (
    dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.q1.io.orc.OrcSerde'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.q1.io.SymlinkTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.q1.io.IgnoreKeyTextOutputFormat'
LOCATION 's3://source-bucket/config-ID/hive/'
TBLPROPERTIES (
```

```

"projection.enabled" = "true",
"projection.dt.type" = "date",
"projection.dt.format" = "yyyy-MM-dd-HH-mm",
"projection.dt.range" = "2022-01-01-00-00,NOW",
"projection.dt.interval" = "1",
"projection.dt.interval.unit" = "HOURS"
);

```

- Wenn Sie Athena verwenden, um einen Parquet -formatierten Inventarbericht abzufragen, verwenden Sie die Beispielabfrage für einen ORC -formatierten Bericht. Verwenden Sie in der Anweisung jedoch den folgenden Parquet SerDe Text anstelle von. ORC SerDe ROW FORMAT SERDE

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'
```

- Wenn Sie Athena verwenden, um einen Inventarbericht im CSV -Format abzufragen, verwenden Sie die folgende Beispielabfrage als Vorlage.

Die folgende Beispielabfrage umfasst alle optionalen Felder in einem Inventarbericht im CSV -Format.

Gehen Sie wie folgt vor, um diese Beispielabfrage zu verwenden:

- Ersetzen Sie *your_table_name* durch den Namen der Athena-Tabelle, die Sie erstellt haben.
- Entfernen Sie alle optionalen Felder, die Sie für Ihren Bestand nicht ausgewählt haben, sodass die Abfrage den für Ihren Bestand gewählten Feldern entspricht.
- Ersetzen Sie den folgenden Bucket-Namen und den Bestandsort (die Konfigurations-ID) entsprechend Ihrer Konfiguration.

```
s3://amzn-s3-demo-bucket/config-ID/hive/
```

- Ersetzen Sie das Datum *2022-01-01-00-00* unter `projection.dt.range` mit dem ersten Tag des Zeitbereichs, innerhalb dessen Sie die Daten in Athena partitionieren. Weitere Informationen finden Sie unter [Partitionieren von Daten in Athena](#).

```

CREATE EXTERNAL TABLE your_table_name(
    bucket string,
    key string,
    version_id string,
    is_latest boolean,
    is_delete_marker boolean,

```

```

        size string,
        last_modified_date string,
        e_tag string,
        storage_class string,
        is_multipart_uploaded boolean,
        replication_status string,
        encryption_status string,
        object_lock_retain_until_date string,
        object_lock_mode string,
        object_lock_legal_hold_status string,
        intelligent_tiering_access_tier string,
        bucket_key_status string,
        checksum_algorithm string,
        object_access_control_list string,
        object_owner string
    ) PARTITIONED BY (
        dt string
    )
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
LOCATION 's3://source-bucket/config-ID/hive/'
TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "HOURS"
);

```

3. Sie können jetzt verschiedene Bestandsabfragen ausführen, wie in den folgenden Beispielen gezeigt. Ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

```

# Get a list of the latest inventory report dates available.
SELECT DISTINCT dt FROM your_table_name ORDER BY 1 DESC limit 10;

# Get the encryption status for a provided report date.
SELECT encryption_status, count(*) FROM your_table_name WHERE dt = 'YYYY-MM-DD-HH-MM' GROUP BY encryption_status;

# Get the encryption status for inventory report dates in the provided range.
SELECT dt, encryption_status, count(*) FROM your_table_name

```



```
WHERE dt > 'YYYY-MM-DD-HH-MM' AND dt < 'YYYY-MM-DD-HH-MM' GROUP BY dt,
encryption_status;
```

Wenn Sie S3 Inventory so konfigurieren, dass das Feld Object Access Control List (ObjectACL) zu einem Inventarbericht hinzugefügt wird, zeigt der Bericht den Wert für das ACL Objektfeld als Base64-kodierte Zeichenfolge an. Um den dekodierten Wert JSON für das ACL Objektfeld zu erhalten, können Sie dieses Feld mit Athena abfragen. Sehen Sie sich die folgenden Abfragebeispiele an. Weitere Informationen zum ACL Objektfeld finden Sie unter [Arbeiten mit dem Feld „Objekt-ACL“](#)

```
# Get the S3 keys that have Object ACL grants with public access.
WITH grants AS (
  SELECT key,
    CAST(
      json_extract(from_utf8(from_base64(object_access_control_list)),
        '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
    ) AS grants_array
  FROM your_table_name
)
SELECT key,
  grants_array,
  grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'uri') = 'http://acs.amazonaws.com/groups/global/AllUsers'
```

```
# Get the S3 keys that have Object ACL grantees in addition to the object owner.
WITH grants AS
  (SELECT key,
    from_utf8(from_base64(object_access_control_list)) AS
    object_access_control_list,
    object_owner,
    CAST(json_extract(from_utf8(from_base64(object_access_control_list)),
      '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))) AS grants_array
  FROM your_table_name)
SELECT key,
  grant,
  objectowner
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE cardinality(grants_array) > 1 AND element_at(grant, 'canonicalId') !=
object_owner;
```

```
# Get the S3 keys with READ permission that is granted in the Object ACL.
WITH grants AS (
  SELECT key,
    CAST(
      json_extract(from_utf8(from_base64(object_access_control_list)),
        '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
    ) AS grants_array
  FROM your_table_name
)
SELECT key,
  grants_array,
  grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'permission') = 'READ';
```

```
# Get the S3 keys that have Object ACL grants to a specific canonical user ID.
WITH grants AS (
  SELECT key,
    CAST(
      json_extract(from_utf8(from_base64(object_access_control_list)),
        '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
    ) AS grants_array
  FROM your_table_name
)
SELECT key,
  grants_array,
  grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'canonicalId') = 'user-canonical-id';
```

```
# Get the number of grantees on the Object ACL.
SELECT key,
  object_access_control_list,
  json_array_length(json_extract(object_access_control_list, '$.grants')) AS
  grants_count
FROM your_table_name;
```

Weitere Informationen zur Verwendung von Athena finden Sie im [Amazon Athena-Benutzerhandbuch](#).

Konvertieren leerer Versions-ID-Strings in Amazon-S3-Inventory-Berichten in Null-Zeichenfolgen

Note

Das folgende Verfahren gilt nur für Amazon S3-Lagerbestandsberichte, die alle Versionen enthalten, und nur, wenn die Berichte „Alle Versionen“ als Manifeste für S3-Batch-Operationen bei Buckets verwendet werden, bei denen S3-Versionierung aktiviert ist. Sie müssen keine Zeichenfolgen für S3-Inventory-Berichte konvertieren, die nur die aktuelle Version angeben.

Sie können S3-Inventory-Berichte als Manifeste für S3-Batchvorgänge verwenden. Wenn jedoch die S3-Versionsverwaltung für einen Bucket aktiviert ist, markieren S3-Inventory-Berichte, die alle Versionen enthalten, alle nullversionierten Objekte mit leeren Zeichenfolgen im Versions-ID-Feld. Wenn ein Inventory-Bericht alle Objektversions-IDs enthält, erkennen Batchvorgänge null-Zeichenfolgen als Versions-IDs, jedoch keine leeren Zeichenfolgen.

Wenn ein S3-Batchvorgangsauftrag einen S3-Inventory-Bericht „alle Versionen“ als Manifest verwendet, schlägt er alle Aufgaben für Objekte fehl, die eine leere Zeichenfolge im Feld Versions-ID haben. Gehen Sie wie folgt vor, um leere Zeichenfolgen im Versions-ID-Feld des S3-Inventory-Berichts in null-Zeichenfolgen für Batchvorgänge zu konvertieren.

Aktualisieren eines Amazon-S3-Inventory-Berichts zur Verwendung mit Batchvorgängen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Navigieren Sie zu Ihrem S3-Inventory-Bericht. Der Inventory-Bericht befindet sich im Ziel-Bucket, den Sie beim Konfigurieren Ihres Inventory-Berichts angegeben haben. Weitere Informationen zum Auffinden von Inventory-Berichten finden Sie unter [Lokalisieren Ihrer Bestandsliste](#).
 - a. Wählen Sie den Ziel-Bucket aus.
 - b. Wählen Sie den Ordner aus. Der Ordner ist nach dem ursprünglichen Quell-Bucket benannt.
 - c. Wählen Sie den nach der Inventory-Konfiguration benannten Ordner aus.

- d. Aktivieren Sie das Kontrollkästchen neben dem Ordner namens Hive aus. Wählen Sie oben auf der Seite S3-URI kopieren aus, um den S3-URI für den Ordner zu kopieren.
3. Öffnen Sie die Amazon-Athena-Konsole unter <https://console.aws.amazon.com/athena/>.
4. Wählen Sie im Abfrage-Editor Einstellungen und anschließend Verwalten aus. Wählen Sie auf der Seite Einstellungen verwalten für Standort des Abfrageergebnisses einen S3-Bucket aus, um Ihre Abfrageergebnisse zu speichern.
5. Erstellen Sie im Abfrage-Editor eine Athena-Tabelle, um die Daten mit dem folgenden Befehl im Inventory-Bericht zu speichern. Ersetzen Sie *table_name* durch einen Namen Ihrer Wahl, und fügen Sie in der LOCATION-Klausel den zuvor kopierten S3-URI ein. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen.

```
CREATE EXTERNAL TABLE table_name(bucket string, key string,  
version_id string) PARTITIONED BY (dt string)ROW FORMAT SERDE  
'org.apache.hadoop.hive.serde2.OpenCSVSerde'STORED AS INPUTFORMAT  
'org.apache.hadoop.hive.q1.io.SymlinkTextInputFormat' OUTPUTFORMAT  
'org.apache.hadoop.hive.q1.io.IgnoreKeyTextOutputFormat' LOCATION 'Copied S3 URI';
```

6. Um den Abfrage-Editor zu löschen, wählen Sie Löschen. Laden Sie den Inventory-Bericht mit dem folgenden Befehl in die Tabelle. Ersetzen Sie *table_name* durch denjenigen, den Sie im vorherigen Schritt ausgewählt haben. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen.

```
MSCK REPAIR TABLE table_name;
```

7. Um den Abfrage-Editor zu löschen, wählen Sie Löschen. Führen Sie die folgende SELECT-Abfrage aus, um alle Einträge im ursprünglichen Inventory-Bericht abzurufen und alle leeren Versions-IDs durch null-Zeichenfolgen zu ersetzen. Ersetzen Sie *table_name* durch das zuvor gewählte, und ersetzen Sie *YYYY-MM-DD-HH-MM* in der WHERE-Klausel durch das Datum des Inventory-Berichts, auf dem dieses Tool ausgeführt werden soll. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen.

```
SELECT bucket as Bucket, key as Key, CASE WHEN version_id = '' THEN 'null' ELSE  
version_id END as VersionId FROM table_name WHERE dt = 'YYYY-MM-DD-HH-MM';
```

8. Kehren Sie zur Amazon-S3-Konsole (<https://console.aws.amazon.com/s3/>) zurück und navigieren Sie zu dem S3-Bucket, den Sie zuvor für den Standort des Abfrageergebnisses ausgewählt haben. Im Inneren sollte es eine Reihe von Ordnern geben, die mit dem Datum enden.

Sie sollten beispielsweise etwas wie `s3://DOC-EXAMPLE-BUCKET/query-result-location/Unsaved/2021/10/07/` sehen. Sie sollten `.csv`-Dateien sehen, die die Ergebnisse der ausgeführten `SELECT`-Abfrage enthalten.

Wählen Sie die CSV-Datei mit dem letzten Änderungsdatum aus. Laden Sie diese Datei für den nächsten Schritt auf Ihren lokalen Computer herunter.

- Die generierte CSV-Datei enthält eine Headerzeile. Um diese CSV-Datei als Eingabe für einen S3-Batchvorgangsauftrag zu verwenden, müssen Sie den Header entfernen, da der Batchvorgang keine Header auf CSV-Manifesten unterstützt.

Um den Header zu entfernen, können Sie einen der folgenden Befehle für die Datei ausführen. Ersetzen Sie `file.csv` durch den Namen von Ihrer CSV-Datei.

Führen Sie auf macOS- und Linux-Computern den `tail`-Befehl in einem Terminalfenster aus.

```
tail -n +2 file.csv > tmp.csv && mv tmp.csv file.csv
```

Führen Sie für Windows-Computer das folgende Skript in einem Windows-PowerShell-Fenster aus. Ersetzen Sie `File-location` durch den Dateipfad und `file.csv` durch den Namen Ihrer Datei.

```
$ins = New-Object System.IO.StreamReader File-location\file.csv
$out = New-Object System.IO.StreamWriter File-location\temp.csv
try {
    $skip = 0
    while ( !$ins.EndOfStream ) {
        $line = $ins.ReadLine();
        if ( $skip -ne 0 ) {
            $out.WriteLine($line);
        } else {
            $skip = 1
        }
    }
} finally {
    $out.Close();
    $ins.Close();
}
Move-Item File-location\temp.csv File-location\file.csv -Force
```

10. Nachdem Sie den Header aus der CSV-Datei entfernt haben, können Sie sie als Manifest in einem S3-Batchvorgangsauftrag verwenden. Laden Sie die CSV-Datei in einen S3-Bucket oder einen Speicherort Ihrer Wahl hoch und erstellen Sie dann einen Batchvorgang-Auftrag mit der CSV-Datei als Manifest.

Weitere Informationen zum Erstellen eines S3-Batchvorgang-Auftrags finden Sie unter [Erstellen eines S3-Batch-Vorgangsauftrags](#).

Arbeiten mit dem Feld „Objekt-ACL“

Eine Amazon-S3-Inventary-Bericht enthält eine Liste der Objekte im S3-Quell-Bucket sowie die Metadaten für jedes Objekt. Das Feld „Objekt-Zugriffsteuerungsliste (ACL)“ ist ein Metadatenfeld, das in Amazon S3 Inventory verfügbar ist. Insbesondere enthält das Feld „Objekt-ACL“ die Zugriffsteuerungsliste für jedes Objekt. Die ACL für ein Objekt definiert, welchen AWS-Konten Gruppen Zugriff auf dieses Objekt gewährt wird und welche Art von Zugriff gewährt wird. Weitere Informationen finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#) und [Amazon-S3-Inventory-Liste](#).

Das Feld „Objekt-ACL“ in Berichten von Amazon S3 Inventory ist im JSON-Format definiert. Die JSON-Daten enthalten die folgenden Felder:

- `version` – Die Version des Objekt-ACL-Feldformats in den Bestandsberichten. Das Datumsformat lautet `yyyy-mm-dd`.
- `status` – Mögliche Werte sind `AVAILABLE` oder `UNAVAILABLE`, um anzugeben, ob eine Objekt-ACL für ein Objekt verfügbar ist. Wenn der Status für die Objekt-ACL `UNAVAILABLE` lautet, lautet der Wert des Felds „Objekteigentümer“ im Bestandsbericht ebenfalls `UNAVAILABLE`.
- `grants` – Berechtigung-Berechtigungsempfänger-Paare, die den Berechtigungsstatus jedes Berechtigungsempfängers auflisten, der von der Objekt-ACL gewährt wird. Die verfügbaren Werte für einen Berechtigungsempfänger sind `CanonicalUser` und `Group`. Weitere Informationen zu den Berechtigungsempfängern finden Sie unter [Berechtigungsempfänger in Zugriffssteuerungslisten](#).

Für einen Berechtigungsempfänger des Typs `Group` umfasst ein Berechtigung-Berechtigungsempfänger-Paar die folgenden Attribute:

- `uri` – Eine vordefinierte Amazon-S3-Gruppe.
- `permission` – Die ACL-Berechtigungen, die für das Objekt erteilt wurden. Weitere Informationen finden Sie unter [ACL-Berechtigungen für ein Objekt](#).

- `type` – Den Typ `Group`, was bedeutet, dass es sich bei dem Berechtigungsempfänger um eine Gruppe handelt.

Für einen Berechtigungsempfänger des Typs `CanonicalUser` umfasst ein Berechtigung-Berechtigungsempfänger-Paar die folgenden Attribute:

- `canonicalId` – Eine verschleierte Form der AWS-Konto -ID. Die kanonische Benutzer-ID für ein AWS-Konto ist spezifisch für dieses Konto. Sie können die kanonische Benutzer-ID abrufen. Weitere Informationen [finden Sie unter Finden Sie die kanonische Benutzer-ID für Sie AWS-Konto](#) im Referenzhandbuch zur AWS Kontoverwaltung.

Note

Wenn es sich bei einem Empfänger in einer ACL um die E-Mail-Adresse eines Empfängers handelt AWS-Konto, verwendet S3 Inventory diese Adresse AWS-Konto und den `canonicalId` `CanonicalUser` Typ, um diesen Empfänger anzugeben. Weitere Informationen finden Sie unter [Berechtigungsempfänger in Zugriffssteuerungslisten](#).

- `permission` – Die ACL-Berechtigungen, die für das Objekt erteilt wurden. Weitere Informationen finden Sie unter [ACL-Berechtigungen für ein Objekt](#).
- `type`— Der Typ `CanonicalUser`, der angibt, dass es sich bei dem Empfänger um einen handelt. AWS-Konto

Das folgende Beispiel zeigt mögliche Werte für das Feld „Objekt-ACL“ im JSON-Format:

```
{
  "version": "2022-11-10",
  "status": "AVAILABLE",
  "grants": [{
    "uri": "http://acs.amazonaws.com/groups/global/AllUsers",
    "permission": "READ",
    "type": "Group"
  }, {
    "canonicalId": "example-canonical-id",
    "permission": "FULL_CONTROL",
    "type": "CanonicalUser"
  }]
}
```

Note

Das Feld „Objekt-ACL“ ist im JSON-Format definiert. In einem Bestandsbericht wird der Wert für das Feld „Objekt-ACL“ als Base64-codierte Zeichenfolge angezeigt.

Angenommen, Sie haben das folgende Feld „Objekt-ACL“ im JSON-Format:

```
{
  "version": "2022-11-10",
  "status": "AVAILABLE",
  "grants": [{
    "canonicalId": "example-canonical-user-ID",
    "type": "CanonicalUser",
    "permission": "READ"
  }]
}
```

Das Feld „Objekt-ACL“ ist codiert und wird als die folgende Base64-codierte Zeichenfolge angezeigt:

```
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIzInN0YXR1cyI6IktFWQU1MQUMRSIsImdyYW50cyI6W3siY2Fub25pY2FsSW
```

Um für das Feld „Objekt-ACL“ den dekodierten Wert in JSON abzurufen, können Sie dieses Feld in Amazon Athena abfragen. Abfragebeispiele finden Sie unter [Abfragen von Amazon S3 Inventory mit Amazon Athena](#).

Überblick über das Replizieren von Objekten

Sie können die Replikation verwenden, um das automatische, asynchrone Kopieren von Objekten zwischen Amazon S3 S3-Buckets zu ermöglichen. Buckets, die für die Objektreplikation konfiguriert sind, können denselben gehören AWS-Konto oder von verschiedenen Konten. Sie können Objekte in einen einzelnen Ziel-Bucket oder in mehrere Ziel-Buckets replizieren. Die Ziel-Buckets können unterschiedlich sein AWS-Regionen oder innerhalb derselben Region wie der Quell-Bucket.

Es gibt zwei Arten der Replikation: Live-Replikation und On-Demand-Replikation.

- Live-Replikation — Verwenden Sie die Live-Replikation, um neue und aktualisierte Objekte automatisch zu replizieren, während sie in den Quell-Bucket geschrieben werden. Bei der Live-Replikation werden keine Objekte repliziert, die vor der Einrichtung der Replikation im Bucket

vorhanden waren. Um Objekte zu replizieren, die vor der Einrichtung der Replikation vorhanden waren, verwenden Sie die On-Demand-Replikation.

- Replikation auf Abruf — Verwenden Sie S3 Batch Replication, um bestehende Objekte bei Bedarf aus dem Quell-Bucket in einen oder mehrere Ziel-Buckets zu replizieren. Weitere Informationen zum Replizieren vorhandener Objekte finden Sie unter [Wann die S3-Batch-Replikation verwendet wird](#).

Es gibt zwei Formen der Live-Replikation: die regionsübergreifende Replikation (CRR) und die regionsübergreifende Replikation (SRR).

- Regionsübergreifende Replikation (CRR) — Sie können CRR damit Objekte in verschiedenen Amazon S3 S3-Buckets replizieren AWS-Regionen. Weitere Informationen zu finden CRR Sie unter [the section called “Verwenden der regionsübergreifenden Replikation”](#).
- Replikation in einer einzelnen Region (SRR) — Sie können sie verwenden SRR, um Objekte zwischen Amazon S3 S3-Buckets im selben zu kopieren AWS-Region. Weitere Informationen zu finden SRR Sie unter [the section called “Verwenden von Replikation innerhalb derselben Region”](#).

Themen

- [Warum sollten Sie Replikation verwenden?](#)
- [Verwenden der regionsübergreifenden Replikation](#)
- [Verwenden von Replikation innerhalb derselben Region](#)
- [Wann sollte die bidirektionale Replikation verwendet werden](#)
- [Wann die S3-Batch-Replikation verwendet wird](#)
- [Workload-Anforderungen und Live-Replikation](#)
- [Was repliziert Amazon S3?](#)
- [Anforderungen und Überlegungen für die Replikation](#)
- [Live-Replikation einrichten](#)
- [Live-Replikation verwalten oder pausieren](#)
- [Replizieren bestehender Objekte mit S3-Batch-Replikation](#)
- [Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status](#)

Warum sollten Sie Replikation verwenden?

Die Replikation unterstützt Sie bei Folgendem:

- Objekte replizieren und dabei Metadaten beibehalten — Sie können die Replikation verwenden, um Kopien Ihrer Objekte zu erstellen, die alle Metadaten, wie z. B. die Erstellungszeit und Version IDs des ursprünglichen Objekts, beibehalten. Diese Funktion ist wichtig, wenn Sie sicherstellen müssen, dass Ihr Replikat mit dem Quellobjekt identisch ist.
- Replikation von Objekten in verschiedene Speicherklassen – Sie können mit der Replikation Objekte direkt in S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive oder eine andere Speicherkategorie in den Ziel-Buckets platzieren. Sie können Ihre Daten auch in dieselbe Speicherkategorie replizieren und Lebenszyklus-Konfigurationen für die Ziel-Buckets verwenden, um Ihre Objekte bei zunehmendem Alter in eine Speicherkategorie für seltener genutzte Objekte zu verschieben.
- Objektkopien unter unterschiedlichen Besitzern verwalten — Unabhängig davon, wem das Quellobjekt gehört, können Sie Amazon S3 anweisen, den Besitz des Replikats auf das AWS-Konto dem der Ziel-Bucket gehört. Diese Instance wird als Eigentümer-Überschreibungs-Option bezeichnet. Sie können diese Option nutzen, um den Zugriff auf Objektreplikate einzuschränken.
- Bewahren Sie Objekte auf mehreren AWS-Regionen— Um sicherzustellen, dass es geografische Unterschiede in Bezug auf den Speicherort Ihrer Daten gibt, können Sie mehrere Ziel-Buckets für verschiedene Bereiche festlegen AWS-Regionen. Diese Funktion kann Ihnen dabei helfen, bestimmte Compliance-Anforderungen zu erfüllen.
- Objekte innerhalb von 15 Minuten replizieren — Um Ihre Daten in derselben zu replizieren AWS-Region Sie können S3 Replication Time Control (S3RTC) verwenden oder innerhalb eines vorhersehbaren Zeitraums über verschiedene Regionen hinweg. S3 RTC repliziert 99,99 Prozent der in Amazon S3 gespeicherten neuen Objekte innerhalb von 15 Minuten (unterstützt durch ein Service Level Agreement). Weitere Informationen finden Sie unter [the section called “Verwenden der S3-Replikationszeitkontrolle”](#).

Note

S3 gilt RTC nicht für die Batch-Replikation. Die Batch-Replikation ist ein On-Demand-Replikationsauftrag und kann mit S3-Batchvorgängen verfolgt werden. Weitere Informationen finden Sie unter [Verfolgen von Auftragsstatus- und Abschluss](#).

- Synchronisieren Sie Buckets, replizieren Sie vorhandene Objekte und replizieren Sie zuvor fehlgeschlagene oder replizierte Objekte – Verwenden Sie die Batch-Replikation als On-Demand-

Replikationsaktion, um Buckets zu synchronisieren und vorhandene Objekte zu replizieren. Weitere Informationen dazu, wann Sie die Batch-Replikation verwenden sollten, finden Sie unter [Wann die S3-Batch-Replikation verwendet wird](#).

- Replizieren Sie Objekte und führen Sie einen Failover zu einem Bucket in einem anderen durch AWS-Region— Um während der Datenreplikation alle Metadaten und Objekte in allen Buckets synchron zu halten, verwenden Sie Regeln für die bidirektionale Replikation (auch als bidirektionale Replikation bezeichnet), bevor Sie die Failover-Steuerung für Amazon S3 Multi-Region Access Point konfigurieren. Bidirektionale Replikationsregeln tragen dazu bei, sicherzustellen, dass wenn Daten in den S3-Bucket geschrieben werden, auf den der Datenverkehr bei einem Failover zurückgreift, diese Daten dann zurück in den Quell-Bucket repliziert werden.

Verwenden der regionsübergreifenden Replikation

S3 Cross-Region Replication (CRR) wird verwendet, um Objekte zwischen Amazon S3 S3-Buckets in verschiedene zu kopieren AWS-Regionen. CRR kann Ihnen helfen, Folgendes zu tun:

- Einhalten der Compliance-Anforderungen – Auch wenn Amazon S3 Ihre Daten standardmäßig in mehreren geografisch entfernten Availability Zones speichert, machen es die Compliance-Anforderungen möglicherweise erforderlich, Daten in noch größeren Entfernungen zu speichern. Um diese Anforderungen zu erfüllen, verwenden Sie die regionsübergreifende Replikation, um Daten zwischen entfernten Standorten zu replizieren AWS-Regionen.
- Latenz minimieren — Wenn sich Ihre Kunden an zwei geografischen Standorten befinden, können Sie die Latenz beim Zugriff auf Objekte minimieren, indem Sie Objektkopien in AWS-Regionen die sich geografisch näher an Ihren Benutzern befinden.
- Steigern Sie die betriebliche Effizienz — wenn Sie Rechencluster in zwei verschiedenen Bereichen haben AWS-Regionen bei denen dieselbe Gruppe von Objekten analysiert wird, können Sie sich dafür entscheiden, Objektkopien in diesen Regionen beizubehalten.

Verwenden von Replikation innerhalb derselben Region

Same-Region-Replication (SRR) wird verwendet, um Objekte zwischen Amazon S3 S3-Buckets in derselben zu kopieren AWS-Region. SRR kann Ihnen helfen, Folgendes zu tun:

- Aggregieren von Protokollen in einen einzelnen Bucket – Wenn Sie Protokolle in mehrere Buckets oder kontoübergreifend speichern, können Sie Protokolle ganz einfach in einen einzelnen Bucket

innerhalb derselben Region replizieren. Dies ermöglicht eine einfachere Protokollverarbeitung an einem einzelnen Standort.

- Konfigurieren von Live-Replikation zwischen Produktions- und Testkonten – Wenn Sie oder Ihre Kunden Produktions- und Testkonten haben, die dieselben Daten nutzen, können Sie Objekte kontoübergreifend replizieren und dabei die Objektmetadaten erhalten.
- Halten Sie sich an die Gesetze zur Datenhoheit — Möglicherweise müssen Sie mehrere Kopien Ihrer Daten getrennt speichern AWS-Konten innerhalb einer bestimmten Region. Mit der Replikation innerhalb derselben Region können Sie automatisch kritische Daten replizieren, wenn Ihre Daten aufgrund von Compliance-Regelungen im Land bleiben müssen.

Wann sollte die bidirektionale Replikation verwendet werden

- Erstellen Sie gemeinsam genutzte Datensätze für mehrere AWS-Regionen— Mit der Replikatmodifikationssynchronisierung können Sie Metadatenänderungen wie Objektzugriffskontrolllisten (ACLs), Objekt-Tags oder Objektsperren auf einfache Weise für Replikationsobjekte replizieren. Diese bidirektionale Replikation ist wichtig, wenn Sie alle Objekte und Objektmetadatenänderungen synchron halten möchten. Sie können die [Synchronisierung von Replikatänderungen für eine neue oder bestehende Replikationsregel aktivieren](#), wenn Sie eine bidirektionale Replikation zwischen zwei oder mehr Buckets in demselben oder unterschiedlichen Buckets durchführen AWS-Regionen.
- Halten Sie die Daten während eines Failovers regionsübergreifend synchron — Sie können Daten in Buckets zwischen AWS-Regionen indem Sie Regeln für die bidirektionale Replikation mit der regionsübergreifenden S3-Replikation (CRR) direkt von einem multiregionalen Access Point aus konfigurieren. Um eine fundierte Entscheidung darüber zu treffen, wann ein Failover eingeleitet werden soll, können Sie auch S3-Replikationsmetriken aktivieren, sodass Sie die Replikation in Amazon CloudWatch, in S3 Replication Time Control (S3RTC) oder vom Multi-Region Access Point aus überwachen können.
- Sorgen Sie für hohe Verfügbarkeit Ihrer Anwendung – Selbst im Fall einer regionalen Datenverkehrsunterbrechung können Sie bidirektionale Replikationsregeln verwenden, um alle Metadaten und Objekte während der Datenreplikation bucketübergreifend synchron zu halten.

Wann die S3-Batch-Replikation verwendet wird

Die Batch-Replikation repliziert vorhandene Objekte als On-Demand-Option in verschiedene Buckets. Im Gegensatz zur Live-Replikation können diese Aufträge nach Bedarf ausgeführt werden. Die Batch-Replikation unterstützt Sie bei Folgendem:

- Replizieren vorhandener Objekte – Sie können die Batch-Replikation verwenden, um Objekte zu replizieren, die dem Bucket hinzugefügt wurden, bevor die Replikation innerhalb derselben Region oder die Replikation in mehreren Regionen konfiguriert wurden.
- Objekte replizieren, die zuvor nicht repliziert werden konnten — Sie können einen Batch-Replikationsauftrag filtern, um zu versuchen, Objekte mit dem Replikationsstatus von zu replizieren. FAILED
- Objekte replizieren, die bereits repliziert wurden — Möglicherweise müssen Sie mehrere Kopien Ihrer Daten getrennt speichern AWS-Konten or AWS-Regionen. Die Batch-Replikation kann vorhandene Objekte an neu hinzugefügte Ziele replizieren.
- Replizieren von Replikaten von Objekten, die aus einer Replikationsregel erstellt wurden – Replikationskonfigurationen erstellen Replikate von Objekten in Ziel-Buckets. Replikate von Objekten können nur mit Batch-Replikation repliziert werden.

Workload-Anforderungen und Live-Replikation

Abhängig von Ihren Workload-Anforderungen eignen sich einige Replikationstypen besser für Ihren Anwendungsfall als andere. Ermitteln Sie anhand der folgenden Tabelle, welche Art von Replikation Sie für Ihre Situation verwenden sollten und ob Sie S3 Replication Time Control (S3RTC) für Ihren Workload verwenden sollten. S3 RTC repliziert 99,99 Prozent der in Amazon S3 gespeicherten neuen Objekte innerhalb von 15 Minuten (unterstützt durch ein Service Level Agreement oder). SLA Weitere Informationen finden Sie unter [the section called “Verwenden der S3-Replikationszeitkontrolle”](#).

Vergleich der Workload-Anforderungen für die Replikation

Workload-Anforderung	S3 RTC (15 MinutenSLA)	Regionsübergreifende Replikation () CRR	Replikation in einer einzelnen Region () SRR
Replizieren Sie Objekte zwischen	Ja	Ja	Ja

Workload-Anforderung	S3 RTC (15 MinutenSLA)	Regionsübergreifende Replikation () CRR	Replikation in einer einzelnen Region () SRR
verschiedenen AWS-Konten			
Replizieren Sie Objekte innerhalb derselben AWS-Region innerhalb von 24-48 Stunden (nicht SLA gesichert)	Nein	Nein	Ja
Replizieren Sie Objekte zwischen verschiedenen AWS-Regionen innerhalb von 24-48 Stunden (nicht SLA gesichert)	Nein	Ja	Nein
Vorhersehbare Replikationszeit: Unterstützt durchSLA, dass 99,9 Prozent der Objekte innerhalb von 15 Minuten repliziert werden	Ja	Nein	Nein

Was repliziert Amazon S3?

Amazon S3 repliziert nur bestimmte Elemente in Buckets, die für die Replikation konfiguriert sind.

Themen

- [Was wird mit Replikationskonfigurationen repliziert?](#)
- [Was wird mit Replikationskonfigurationen nicht repliziert?](#)
- [So wirkt sich die Standard-Bucket-Verschlüsselung auf die Replikation aus](#)

Was wird mit Replikationskonfigurationen repliziert?

Standardmäßig repliziert Amazon S3 Folgendes:

- Objekte, die nach dem Hinzufügen einer Replikations-Konfiguration erstellt wurden.
- Unverschlüsselte Objekte.
- Objekte, die mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verschlüsselt wurden, Objekte, die im Ruhezustand unter einem von Amazon S3 verwalteten Schlüssel (SSE-S3) verschlüsselt wurden, oder einem KMS Schlüssel, der in gespeichert ist AWS Key Management Service (SSE-KMS). Weitere Informationen finden Sie unter [the section called “Replizieren verschlüsselter Objekte”](#).
- Objektmetadaten von den Quellobjekten zu den Replikaten Informationen zum Replizieren von Metadaten aus den Replikaten zu den Quellobjekten finden Sie unter [Replizieren von Metadatenänderungen mit der Synchronisierung von Amazon-S3-Replikatänderungen](#).
- Nur Objekte im Quell-Bucket, für die der Bucket-Besitzer berechtigt ist, Objekte und Zugriffskontrolllisten (ACLs) zu lesen.

Weitere Informationen zum Eigentum an Ressourcen finden Sie unter [Amazon-S3-Bucket- und Objekt-Eigentümerschaft](#).

- ACL-Objektaktualisierungen, sofern Sie Amazon S3 nicht anweisen, den Besitz des Replikats zu ändern, wenn Quell- und Ziel-Buckets nicht denselben Konten gehören.

Weitere Informationen finden Sie unter [Ändern des Replikat-Eigentümers](#).

Es kann eine Weile dauern, bis Amazon S3 die beiden ACLs synchronisieren kann. Diese Änderung der Eigentümerschaft gilt nur für Objekte, die erstellt wurden, nachdem Sie dem Bucket eine Replikationskonfiguration hinzugefügt haben.

- Objekt-Markierungen, sofern vorhanden.
- Informationen zur Beibehaltung der S3-Objektsperre, sofern vorhanden.

Wenn Amazon S3 Objekte mit angewandten Aufbewahrungsinformationen repliziert, wendet es dieselben Aufbewahrungssteuerungen auf Ihre Replikate an, wodurch der für Ihre Ziel-Buckets konfigurierte standardmäßige Aufbewahrungszeitraum überschrieben wird. Wenn auf die Objekte in Ihrem Quell-Bucket keine Aufbewahrungssteuerungen angewandt sind und Sie sie in Ziel-Buckets mit festgelegtem Standard-Aufbewahrungszeitraum replizieren, wird der Standard-Aufbewahrungszeitraum der Ziel-Buckets auf Ihre Objektreplikate angewandt. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Auswirkungen von Löschvorgängen auf die Replikation

Wenn Sie ein Objekt aus dem Quell-Bucket löschen, werden standardmäßig die folgenden Aktionen ausgeführt:

- Wenn Sie eine DELETE Anfrage stellen, ohne eine Objektversions-ID anzugeben, fügt Amazon S3 eine Löschmarkierung hinzu. Amazon S3 befasst sich wie folgt mit der Löschmarkierung:
 - Wenn Sie die aktuelle Version der Replikations-Konfiguration verwenden, d. h. das Element `Filter` in einer Replikations-Konfigurations-Regel angeben, repliziert Amazon S3 die Löschmarkierung nicht standardmäßig. Sie können den non-tag-based Regeln jedoch die Replikation von Löschmarkierungen hinzufügen. Weitere Informationen finden Sie unter [Replizieren von Löschmarkierungen auf Buckets](#).
 - Wenn Sie das Element `Filter` nicht angeben, geht Amazon S3 davon aus, dass die Replikations-Konfiguration Version V1 ist, und repliziert Löschmarkierungen, die aus Benutzeraktionen resultierten. Wenn Amazon S3 jedoch ein Objekt aufgrund einer Lebenszyklus-Aktion löscht, wird die Löschmarkierung nicht auf die Ziel-Buckets repliziert.
- Wenn Sie angeben, dass eine Objektversions-ID in einer DELETE-Anforderung gelöscht werden soll, löscht Amazon S3 diese Objektversion im Quell-Bucket. Die Löschung wird jedoch nicht in den Ziel-Buckets repliziert. Anders ausgedrückt: Dieselbe Objektversion wird aus den Ziel-Buckets nicht gelöscht. Dies schützt Daten vor missbräuchlichen Löschungen.

Was wird mit Replikationskonfigurationen nicht repliziert?

Standardmäßig repliziert Amazon S3 Folgendes nicht:

- Objekte im Quell-Bucket, bei denen es sich um Replikate handelt, die von einer anderen Replikationsregel erstellt wurden. Zum Beispiel: Angenommen Sie konfigurieren eine Replikation, bei der Bucket A die Quelle und Bucket B das Ziel ist. Nehmen wir jetzt an, Sie fügen eine weitere Replikations-Konfiguration hinzu, bei der Bucket B die Quelle und Bucket C das Ziel ist. In diesem Fall werden Objekte in Bucket B, die Replikate von Objekten in Bucket A sind, nicht in Bucket C repliziert.

Verwenden Sie die Batch-Replikation, um Objekte zu replizieren, die Replikate sind. Weitere Informationen zum Konfigurieren der Batch-Replikation finden Sie unter [Replizieren vorhandener Objekte](#).

- Objekte im Quell-Bucket, die bereits auf ein anderes Ziel repliziert wurden. Wenn Sie beispielsweise den Ziel-Bucket in einer vorhandenen Replikationskonfiguration ändern, repliziert Amazon S3 diese Objekte nicht erneut.

Verwenden Sie die Batch-Replikation, um zuvor replizierte Objekte zu replizieren. Weitere Informationen zum Konfigurieren der Batch-Replikation finden Sie unter [Replizieren vorhandener Objekte](#).

- Die Batch-Replikation unterstützt nicht das erneute Replizieren von Objekten, die mit der Versions-ID des Objekts aus dem Ziel-Bucket gelöscht wurden. Wenn Sie diese Objekte erneut replizieren möchten, können Sie die Quellobjekte mit einem Batch-Kopierauftrag kopieren. Beim Kopieren dieser Objekte werden neue Versionen des Objekts im Quell-Bucket erstellt und die Replikation zum Ziel wird automatisch initiiert. Weitere Informationen zur Verwendung von Batch Copy finden Sie unter [Beispiele, die Batch-Vorgänge zum Kopieren von Objekten verwenden](#).
- Standardmäßig beim Replizieren von einem anderen AWS-Konto, zum Quell-Bucket hinzugefügte Löschmarkierungen werden nicht repliziert.

Informationen zum Replizieren von Löschmarkierungen finden Sie unter [Replizieren von Löschmarkierungen auf Buckets](#).

- Objekte, die in den Speicherklassen oder Stufen S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access oder S3 Intelligent-Tiering Deep Archive Access gespeichert sind. Sie können diese Objekte erst replizieren, wenn Sie sie wiederhergestellt und in eine andere Speicherklasse kopiert haben.

Weitere Informationen zu S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive finden Sie unter [Speicherklassen für Objekte, auf die selten zugegriffen wird](#).

Weitere Informationen über S3 Intelligent-Tiering finden Sie unter [Amazon S3 Intelligent Tiering](#)

- Objekte im Quell-Bucket, für die der Bucket-Eigentümer nicht über ausreichende Berechtigungen zur Replikation verfügt.

Weitere Informationen darüber, wie ein Objekt-Eigentümer einem Bucket-Eigentümer Berechtigungen erteilen kann, finden Sie unter [Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt](#).

- Aktualisierungen von Unterressourcen auf Bucket-Ebene.

Wenn Sie beispielsweise die Lebenszyklus-Konfiguration ändern oder eine Benachrichtigungskonfiguration zu Ihrem Quell-Bucket hinzufügen, werden diese Änderungen nicht auf den Ziel-Bucket angewendet. Durch diese Funktion ist es möglich, für den Quell- und den Ziel-Bucket verschiedene Konfigurationen zu nutzen.

- Aktionen, die von der Lebenszyklus-Konfiguration durchgeführt werden.

Wenn eine Lebenszyklus-Konfiguration beispielsweise nur auf Ihrem Quell-Bucket aktiviert ist, erstellt Amazon S3 Löschmarkierungen für abgelaufene Objekte, repliziert diese Markierungen jedoch nicht. Wenn Sie dieselbe Lebenszyklus-Konfiguration sowohl auf den Quell- als auch auf den Ziel-Bucket anwenden möchten, aktivieren Sie für beide Buckets dieselbe Lebenszyklus-Konfiguration. Weitere Informationen zur Lebenszyklus-Konfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

- Wenn Sie tagbasierte Replikationsregeln mit Live-Replikation verwenden, müssen neue Objekte während des Vorgangs mit dem entsprechenden Tag für die Replikationsregel gekennzeichnet werden. PutObject Andernfalls werden die Objekte nicht repliziert. Wenn Objekte nach dem PutObject Vorgang markiert werden, werden diese Objekte ebenfalls nicht repliziert.

Um Objekte zu replizieren, die nach dem PutObject Vorgang markiert wurden, müssen Sie S3 Batch Replication verwenden. Weitere Informationen zur Batch-Replikationen finden Sie unter [Replizieren vorhandener Objekte](#).

So wirkt sich die Standard-Bucket-Verschlüsselung auf die Replikation aus

Wenn Sie die Standard-Verschlüsselung für einen Replikations-Ziel-Bucket aktivieren, gilt das folgende Verschlüsselungsverhalten:

- Wenn Objekte im Quell-Bucket nicht verschlüsselt sind, werden die Replikatobjekte im Ziel-Bucket mithilfe der Einstellungen der Standard-Verschlüsselung des Ziel-Buckets verschlüsselt. Infolgedessen unterscheiden sich die Entity-Tags (ETags) der Quellobjekte von denen ETags der Replikatobjekte. Wenn Sie Anwendungen haben, die dies verwenden ETags, müssen Sie diese Anwendungen aktualisieren, um diesem Unterschied Rechnung zu tragen.
- Wenn Objekte im Quell-Bucket mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt werden, serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS), die Replikatobjekte im Ziel-Bucket

verwenden denselben Verschlüsselungstyp wie die Quellobjekte. Die Einstellungen der Standard-Verschlüsselung des Ziel-Buckets werden nicht verwendet.

Anforderungen und Überlegungen für die Replikation

Für die Amazon S3 S3-Replikation ist Folgendes erforderlich:

- Der Besitzer des Quell-Buckets muss die Quelle und das Ziel kennen AWS-Regionen für ihr Konto aktiviert. Für das Konto des Ziel-Bucket-Eigentümers muss die Zielregion aktiviert sein.

Weitere Informationen zur Aktivierung oder Deaktivierung eines AWS-Region, siehe [Verwalten AWS-Regionen](#) in der Allgemeine AWS-Referenz.

- Für Quell- und Ziel-Buckets muss die Versioning aktiviert sein. Weitere Informationen über das Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).
- Amazon S3 muss berechtigt sein, Objekte aus dem Quell-Bucket in Ihrem Namen in den Ziel-Bucket/die Ziel-Buckets zu replizieren. Weitere Informationen zu diesen Berechtigungen finden Sie unter [Berechtigungen für die Live-Replikation einrichten](#).
- Wenn der Besitzer des Quell-Buckets nicht Eigentümer des Objekts im Bucket ist, muss der Objekteigentümer dem Bucket-Besitzer READ die READ_ACP entsprechenden Berechtigungen mit der Objektzugriffskontrollliste (ACL) gewähren. Weitere Informationen finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).
- Wenn für den Quell-Bucket die S3-Objektsperre aktiviert ist, muss sie für die Ziel-Buckets ebenfalls aktiviert sein.

Um die Replikation für einen Bucket zu aktivieren, für den Object Lock aktiviert ist, müssen Sie den AWS Command Line Interface, RESTAPI, oder AWS SDKs. Weitere allgemeine Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Note

Sie müssen zwei neue Berechtigungen für den S3-Quell-Bucket im AWS Identity and Access Management (IAM) Rolle, mit der Sie die Replikation einrichten. Die zwei neuen Berechtigungen lauten `s3:GetObjectRetention` und `s3:GetObjectLegalHold`. Wenn die Rolle über eine `s3:Get*`-Berechtigung verfügt, ist die Anforderung erfüllt. Weitere Informationen finden Sie unter [Berechtigungen für die Live-Replikation einrichten](#).

Weitere Informationen finden Sie unter [Live-Replikation einrichten](#).

Wenn Sie die Replikationskonfiguration in einem kontenübergreifenden Szenario einrichten, in dem die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, gilt die folgende zusätzliche Anforderung:

- Der Eigentümer der Ziel-Buckets muss dem Eigentümer des Quell-Buckets mit einer Bucket-Richtlinie die Berechtigungen zum Replizieren von Objekten erteilen. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten](#).
- Die Ziel-Buckets dürfen nicht als Buckets des Typs "Zahlung durch den Anforderer" konfiguriert sein. Weitere Informationen finden Sie unter [Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung](#).

Überlegungen zur Replikation

Bevor Sie eine Replikationskonfiguration erstellen, sollten Sie die folgenden Überlegungen beachten.

Themen

- [Lebenszyklus-Konfiguration und Objektreplicate](#)
- [Versioning-Konfiguration und Replikations-Konfiguration](#)
- [Verwenden der S3-Replikation mit S3-Intelligent-Tiering](#)
- [Protokollierungskonfiguration und Replikations-Konfiguration](#)
- [CRRund die Zielregion](#)
- [S3-Batch-Replikation](#)
- [Steuerung der S3-Replikationszeit](#)

Lebenszyklus-Konfiguration und Objektreplicate

Die von Amazon S3 für die Replikation eines Objekts benötigte Zeit hängt von der Größe des Objekts ab. Bei großen Objekten kann dies mehrere Stunden dauern. Auch wenn es möglicherweise etwas dauert, bis ein Replikat im Ziel verfügbar ist, entspricht die Erstellungszeit des Replikats der Erstellungszeit des entsprechenden Objekts im Quell-Bucket. Wenn eine Lebenszyklus-Konfiguration für einen Ziel-Bucket aktiviert ist, berücksichtigen die Lebenszyklusregeln die ursprüngliche Erstellungszeit des Objekts, nicht den Zeitpunkt, zu dem das Replikat im Ziel-Bucket verfügbar wurde.

Die Replikations-Konfiguration erfordert einen Bucket mit aktiviertem Versioning. Berücksichtigen Sie bei der Aktivierung des Versioning in einem Bucket Folgendes:

- Wenn eine Lebenszyklus-Konfiguration für den Ablauf von Objekten vorhanden ist, nachdem Sie die Versionsverwaltung aktiviert haben, fügen Sie eine `NonCurrentVersionExpiration`-Richtlinie hinzu, damit dasselbe Verhalten hinsichtlich einer dauerhaften Löschung beibehalten wird wie vor der Aktivierung der Versionsverwaltung.
- Wenn eine Übergangs-Lebenszyklus-Konfiguration vorhanden ist, nachdem Sie die Versionsverwaltung aktiviert haben, sollten Sie in Betracht ziehen, eine `NonCurrentVersionTransition`-Richtlinie hinzuzufügen.

Versioning-Konfiguration und Replikations-Konfiguration

Wenn Sie die Replikation auf einem Bucket konfigurieren, muss sowohl für den Quell- als auch für den Ziel-Bucket das Versioning aktiviert sein. Nachdem Sie sowohl im Quell- als auch im Ziel-Bucket das Versioning aktiviert und die Replikation im Quell-Bucket konfiguriert haben, werden Sie folgende Schwierigkeiten beobachten:

- Wenn Sie versuchen, das Versioning auf dem Quell-Bucket zu deaktivieren, gibt Amazon S3 einen Fehler zurück. Sie müssen die Replikations-Konfiguration entfernen, damit Sie das Versioning auf dem Quell-Bucket deaktivieren können.
- Wenn Sie das Versioning auf dem Ziel-Bucket deaktivieren, schlägt die Replikation fehl. Das Quellobjekt weist den Replikationsstatus `FAILED` auf.

Verwenden der S3-Replikation mit S3-Intelligent-Tiering

S3-Intelligent-Tiering ist eine Speicherklasse, die entworfen wurde, um die Speicherkosten zu optimieren, indem sie Daten automatisch zur kostengünstigsten Zugriffsstufe verschiebt. Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt die Objekte, auf die nicht zugegriffen wurde, automatisch in kostengünstigere Zugriffsebenen.

Replizieren von in S3 Intelligent-Tiering gespeicherten Objekten mit S3 Batch Replication oder Aufrufen [CopyObject](#) oder [UploadPartCopy](#) stellt Zugriff dar. In diesen Fällen werden die Quellobjekte der Kopier- oder Replikationsvorgänge hochgestuft.

Weitere Informationen zu S3-Intelligent-Tiering finden Sie unter [Amazon S3 Intelligent Tiering](#).

Protokollierungskonfiguration und Replikations-Konfiguration

Wenn Amazon S3 Protokolle an einen Bucket übermittelt, für den die Replikation aktiviert ist, repliziert der Service die Protokollobjekte.

Wenn Serverzugriffsprotokolle ([Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#)) oder AWS CloudTrail logs ([Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#)) sind in Ihrem Quell- oder Ziel-Bucket aktiviert. Amazon S3 nimmt replikationsbezogene Anfragen in die Protokolle auf. Beispiel: Amazon S3 protokolliert jedes Objekt, das es repliziert.

CRRund die Zielregion

Amazon S3 Cross-Region Replication (CRR) wird verwendet, um Objekte zwischen S3-Buckets in verschiedenen S3-Buckets zu kopieren AWS-Regionen. Sie können die Region für Ihren Ziel-Bucket entweder auf der Grundlage Ihrer Geschäftsanforderungen oder aufgrund von Kostenüberlegungen auswählen. Beispielsweise variieren die Gebühren für eine regionsübergreifende Datenübertragung je nach gewählten Regionen.

Angenommen, Sie wählen USA Ost (Nord-Virginia) (us-east-1) als Region für Ihren Quell-Bucket aus. Wenn Sie USA West (Oregon) (us-west-2) als Region für Ihre Ziel-Buckets auswählen, zahlen Sie mehr, als wenn Sie die Region USA Ost (Ohio) (us-east-2) auswählen. Weitere Informationen zu Preisen finden Sie im Abschnitt „Datenübertragungspreise“ unter [Amazon S3 – Preise](#).

Für die Replikation in derselben Region fallen keine Datenübertragungsgebühren an (SRR).

S3-Batch-Replikation

Informationen zu Überlegungen zur Batch-Replikation finden Sie unter [Überlegungen zur S3-Batch-Replikation](#).

Steuerung der S3-Replikationszeit

Informationen zu bewährten Methoden und Überlegungen zu S3 Replication Time Control (S3RTC) finden Sie unter [Bewährte Methoden und Richtlinien für S3 RTC](#).

Live-Replikation einrichten

Note

Objekte, die vor dem Einrichten der Replikation vorhanden waren, werden nicht automatisch repliziert. Anders ausgedrückt: Amazon S3 repliziert Objekte nicht rückwirkend. Verwenden Sie die S3-Batch-Replikation, um Objekte zu replizieren, die vor Ihrer

Replikationskonfiguration erstellt wurden. Weitere Informationen zum Konfigurieren der Batch-Replikation finden Sie unter [Replizieren vorhandener Objekte](#).

Um die Live-Replikation — Replikation in derselben Region (SRR) oder regionsübergreifende Replikation () — zu aktivieren, fügen Sie Ihrem CRR Quell-Bucket eine Replikationskonfiguration hinzu. Diese Konfiguration weist Amazon S3 an, Objekte wie angegeben zu replizieren. In der Replikations-Konfiguration müssen Sie Folgendes angeben:

- Die Ziel-Buckets – Der Bucket oder die Buckets, in den/die Amazon S3 die Objekte replizieren soll.
- Die Objekte, die Sie replizieren möchten – Sie können alle Objekte im Quell-Bucket replizieren oder nur eine Teilmenge davon. Teilmengen identifizieren Sie, indem Sie ein [Schlüsselnamenpräfix](#), mindestens ein Objekt-Tag oder beides in der Konfiguration angeben.

Wenn Sie beispielsweise eine Replikationsregel konfigurieren, um nur Objekte mit dem Schlüsselnamenpräfix `Tax/` zu replizieren, repliziert Amazon S3 Objekte mit Schlüsseln wie `Tax/doc1` oder `Tax/doc2`. Es repliziert aber keine Objekte mit dem Schlüssel `Lega1/doc3`. Wenn Sie sowohl ein Präfix als auch mindestens ein Tag angeben, repliziert Amazon S3 nur Objekte, die dieses Schlüsselpräfix und diese Tags aufweisen.

- Ein AWS Identity and Access Management (IAM) Rolle — Amazon S3 übernimmt diese IAM Rolle, um Objekte in Ihrem Namen zu replizieren.

Zusätzlich zu diesen Mindestanforderungen können Sie die folgenden Optionen auswählen:

- Replikat-Speicherklasse – Standardmäßig speichert Amazon S3 Objektreplikate in derselben Speicherklasse wie das Quellobjekt. Sie können eine andere Speicherklasse für die Replikate festlegen.
- Replikat-Eigentümerschaft – Amazon S3 geht davon aus, dass ein Objektreplikat weiterhin das Eigentum des Eigentümers des Quellobjekts bleibt. Wenn es also Objekte repliziert, repliziert es auch die entsprechende Objekt-Zugriffskontrollliste (ACL) oder die S3-Einstellung Object Ownership. Wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, können Sie die Replikation so konfigurieren, dass der Besitzer eines Replikats in den AWS-Konto dem der Ziel-Bucket gehört.

Sie können die Replikation konfigurieren RESTAPI, indem Sie AWS SDKs, AWS Command Line Interface (AWS CLI) oder die Amazon S3 S3-Konsole.

Amazon S3 bietet auch API Funktionen zur Unterstützung der Einrichtung von Replikationsregeln. Weitere Informationen finden Sie unter den folgenden Themen in der Amazon Simple Storage Service API Reference:

- [PutBucketReplication](#)
- [GetBucketReplication](#)
- [DeleteBucketReplication](#)

Themen

- [Replikationskonfiguration](#)
- [Berechtigungen für die Live-Replikation einrichten](#)
- [Beispiele für die Konfiguration der Live-Replikation](#)

Replikationskonfiguration

Amazon S3 speichert eine Replikationskonfiguration als XML. In der XML Konfigurationsdatei für die Replikation geben Sie eine an AWS Identity and Access Management (IAM) Rolle und eine oder mehrere Regeln.

```
<ReplicationConfiguration>
  <Role>IAM-role-ARN</Role>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
  ...
</ReplicationConfiguration>
```

Amazon S3 kann ohne Berechtigung durch Sie keine Objekte replizieren. Sie gewähren Berechtigungen mit der IAM Rolle, die Sie in der Replikationskonfiguration angeben. Amazon S3 übernimmt die IAM Rolle, Objekte in Ihrem Namen zu replizieren. Sie müssen der IAM Rolle zunächst die erforderlichen Berechtigungen erteilen. Weitere Informationen zum Verwalten von Berechtigungen finden Sie unter [Berechtigungen für die Live-Replikation einrichten](#).

In den folgenden Szenarien fügen Sie eine Regel zur Replikationskonfiguration hinzu:

- Sie möchten alle Objekte replizieren.
- Sie möchten eine Teilmenge der Objekte replizieren. Sie identifizieren die Teilmenge der Objekte, indem Sie einen Filter zur Regel hinzufügen. In dem Filter geben Sie ein Objektschlüsselpräfix, Markierungen oder eine Kombination aus beidem an, um die Objektteilmenge zu identifizieren, für die die Regel gilt. Die Filter zielen auf Objekte ab, die genau den von Ihnen angegebenen Werten entsprechen.

Sie fügen mehrere Regeln zu einer Replikationskonfiguration hinzu, wenn Sie eine andere Teilmenge von Objekten replizieren möchten. In jeder Regel geben Sie einen Filter an, der eine andere Teilmenge von Objekten auswählt. Beispiel: Sie möchten Objekte mit dem Schlüsselpräfix `tax/` oder `document/` replizieren. Dazu fügen Sie zwei Regeln hinzu, eine, die den `tax/`-Schlüsselpräfix-Filter angibt und eine andere, die das `document/`-Schlüsselpräfix angibt. Weitere Hinweise zum Präfix für Objektschlüssel finden Sie unter [Organisieren von Objekten mit Präfixen](#).

In den folgenden Abschnitten finden Sie zusätzliche Informationen.

Themen

- [Basisregelkonfiguration](#)
- [Optional: Festlegen eines Filters](#)
- [Zusätzliche Zielkonfigurationen](#)
- [Beispiele für Replikations-Konfigurationen](#)
- [Abwärtskompatibilität](#)

Basisregelkonfiguration

Jede Regel muss den Status und die Priorität der Regel enthalten. Die Regel muss auch angeben, ob Löschmarkierungen repliziert werden sollen.

- `Status` gibt an, ob die Regel mithilfe der Werte `Enabled` oder `Disabled` aktiviert oder deaktiviert ist. Wenn eine Regel deaktiviert ist, führt Amazon S3 die in der Regel angegebenen Aktionen nicht durch.
- `Priority` gibt an, welche Regel Vorrang hat, wenn zwei oder mehr Replikationsregeln in Konflikt stehen. Amazon S3 versucht, Objekte gemäß allen Replikationsregeln zu replizieren. Wenn es jedoch zwei oder mehr Regeln mit demselben Ziel-Bucket gibt, werden Objekte gemäß der Regel mit der höchsten Priorität repliziert. Je höher die Zahl, desto höher die Priorität.

- `DeleteMarkerReplication` gibt an, ob Löschmarkierungen unter Verwendung der Werte `Enabled` oder `Disabled` repliziert werden sollen.

In der Zielkonfiguration müssen Sie den Namen des/der Buckets angeben, in den/die Amazon S3 Objekte replizieren soll.

Das folgende Beispiel zeigt die Mindestanforderungen für eine V2-Regel. Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin das XML V1-Format. Weitere Informationen finden Sie unter [Abwärtskompatibilität](#).

```
...
  <Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
      <Status>Enabled-or-Disabled</Status>
    </DeleteMarkerReplication>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
    </Destination>
  </Rule>
  <Rule>
    ...
  </Rule>
  ...
...
```

Sie können auch andere Konfigurationsoptionen festlegen. Beispiel: Sie möchten für Objektreplicate eine andere Speicherklasse verwenden als für das Quellobjekt.

Optional: Festlegen eines Filters

Um eine Teilmenge von Objekten auszuwählen, für die die Regel gilt, fügen Sie einen optionalen Filter hinzu. Sie können nach Objekt-Schlüsselpräfix, Objekt-Tags oder einer Kombination aus beidem filtern. Wenn Sie sowohl nach Schlüsselpräfix als auch nach Objekt-Tags filtern, kombiniert Amazon S3 die Filter mit einem logischen AND-Operator. Anders ausgedrückt: Die Regel wird auf

eine Teilmenge von Objekten mit einem bestimmten Schlüsselpräfix und bestimmten Markierungen angewendet.

Filter basierend auf Objektschlüsselpräfix

Um eine Regel mit einem Filter nach Objektschlüsselpräfix festzulegen, verwenden Sie den folgenden Code. Sie können nur ein Präfix angeben.

```
<Rule>
  ...
  <Filter>
    <Prefix>key-prefix</Prefix>
  </Filter>
  ...
</Rule>
...
```

Filter basierend auf Objekt-Markierungen

Um eine Regel mit einem Filter nach Objekt-Markierungen festzulegen, verwenden Sie den folgenden Code. Sie können einen oder mehrere Objekt-Markierungen festlegen.

```
<Rule>
  ...
  <Filter>
    <And>
      <Tag>
        <Key>key1</Key>
        <Value>value1</Value>
      </Tag>
      <Tag>
        <Key>key2</Key>
        <Value>value2</Value>
      </Tag>
      ...
    </And>
  </Filter>
  ...
</Rule>
...
```

Filtern mit einem Schlüsselpräfix und Objekt-Markierungen

Um eine Regel mit einem Filter nach einer Kombination aus Schlüsselpräfix und Objekt-Markierungen festzulegen, verwenden Sie den folgenden Code. Sie wickeln diese Filter in ein übergeordnetes `<And>`-Element ein. Amazon S3 führt eine logische AND-Operation aus, um diese Filter zu kombinieren. Anders ausgedrückt: Die Regel wird auf eine Teilmenge von Objekten sowohl mit einem bestimmten Schlüsselpräfix als auch mit bestimmten Tags angewendet.

```
<Rule>
  ...
  <Filter>
    <And>
      <Prefix>key-prefix</Prefix>
      <Tag>
        <Key>key1</Key>
        <Value>value1</Value>
      </Tag>
      <Tag>
        <Key>key2</Key>
        <Value>value2</Value>
      </Tag>
      ...
    </Filter>
    ...
  </Rule>
  ...
```

Note

- Wenn Sie eine Regel mit einem leeren `<Filter>` Element angeben, gilt Ihre Regel für alle Objekte in Ihrem Bucket.
- Wenn Sie tagbasierte Replikationsregeln mit Live-Replikation verwenden, müssen neue Objekte während des `PutObject` Vorgangs mit dem entsprechenden Tag für die Replikationsregel gekennzeichnet werden. Andernfalls werden die Objekte nicht repliziert. Wenn Objekte nach dem `PutObject` Vorgang markiert werden, werden diese Objekte ebenfalls nicht repliziert.

Um Objekte zu replizieren, die nach dem `PutObject` Vorgang markiert wurden, müssen Sie S3 Batch Replication verwenden. Weitere Informationen zur Batch-Replikationen finden Sie unter [Replizieren vorhandener Objekte](#).

Zusätzliche Zielkonfigurationen

In der Zielkonfiguration geben Sie den/die Bucket(s) an, in den/die Amazon S3 Objekte replizieren soll. Sie können Konfigurationen einrichten, um Objekte aus einem Quell-Bucket in einen oder mehrere Ziel-Buckets zu replizieren.

```
...
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
</Destination>
...
```

Sie können die folgenden Optionen im Element `<Destination>` hinzufügen.

Themen

- [Festlegen der Speicherklasse](#)
- [Hinzufügen mehrerer Ziel-Buckets](#)
- [Angaben verschiedener Parameter für jede Replikationsregel mit mehreren Ziel-Buckets](#)
- [Ändern der Replikat-Eigentümerschaft](#)
- [Aktivieren der S3 Replication Time Control](#)
- [Replizieren Sie Objekte, die mit serverseitiger Verschlüsselung erstellt wurden, mithilfe von AWS KMS](#)

Festlegen der Speicherklasse

Sie können die Speicherklasse für die Objektreplicate festlegen. Standardmäßig verwendet Amazon S3 wie im folgenden Beispiel die Speicherklasse des Quellobjekts zum Erstellen von Objektreplicaten.

```
...
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
  <StorageClass>storage-class</StorageClass>
</Destination>
...
```

Hinzufügen mehrerer Ziel-Buckets

Sie können wie folgt mehrere Ziel-Buckets in einer einzigen Replikations-Konfiguration hinzufügen.

```

...
<Rule>
  <ID>Rule-1</ID>
  <Status>Enabled-or-Disabled</Status>
  <Priority>integer</Priority>
  <DeleteMarkerReplication>
    <Status>Enabled-or-Disabled</Status>
  </DeleteMarkerReplication>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
  </Destination>
</Rule>
<Rule>
  <ID>Rule-2</ID>
  <Status>Enabled-or-Disabled</Status>
  <Priority>integer</Priority>
  <DeleteMarkerReplication>
    <Status>Enabled-or-Disabled</Status>
  </DeleteMarkerReplication>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET2</Bucket>
  </Destination>
</Rule>
...

```

Angaben verschiedener Parameter für jede Replikationsregel mit mehreren Ziel-Buckets

Wenn Sie mehrere Ziel-Buckets in einer einzigen Replikations-Konfiguration hinzufügen, können Sie wie folgt verschiedene Parameter für jede Replizierungsregel angeben.

```

...
<Rule>
  <ID>Rule-1</ID>
  <Status>Enabled-or-Disabled</Status>
  <Priority>integer</Priority>
  <DeleteMarkerReplication>
    <Status>Disabled</Status>
  </DeleteMarkerReplication>
  <Metrics>
  <Status>Enabled</Status>
  <EventThreshold>
    <Minutes>15</Minutes>
  </EventThreshold>

```

```

</Metrics>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
  </Destination>
</Rule>
<Rule>
  <ID>Rule-2</ID>
  <Status>Enabled-or-Disabled</Status>
  <Priority>integer</Priority>
  <DeleteMarkerReplication>
    <Status>Enabled</Status>
  </DeleteMarkerReplication>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET2</Bucket>
  </Destination>
</Rule>
...

```

Ändern der Replikat-Eigentümerschaft

Wenn die Quell- und Ziel-Buckets nicht denselben Konten gehören, können Sie die Eigentümerschaft des Replikats auf Folgendes ändern AWS-Konto dem der Ziel-Bucket gehört. Fügen Sie dazu das `AccessControlTranslation`-Element hinzu. Dieses Element übernimmt den Wert `Destination`.

```

...
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
  <Account>destination-bucket-owner-account-id</Account>
  <AccessControlTranslation>
    <Owner>Destination</Owner>

```

```

    </AccessControlTranslation>
  </Destination>
  ...

```

Wenn Sie das `AccessControlTranslation` Element nicht zur Replikationskonfiguration hinzufügen, gehören die Replikate demselben AWS-Konto dem das Quellobjekt gehört. Weitere Informationen finden Sie unter [Ändern des Replikat-Eigentümers](#).

Aktivieren der S3 Replication Time Control

Sie können S3 Replication Time Control (S3RTC) in Ihrer Replikationskonfiguration aktivieren. S3 RTC repliziert die meisten Objekte innerhalb von Sekunden und 99,99 Prozent der Objekte innerhalb von 15 Minuten (unterstützt durch ein Service Level Agreement).

Note

Nur ein Wert von `<Minutes>15</Minutes>` wird für `EventThreshold` und `Time` akzeptiert.

```

...
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
</Destination>
...

```

Weitere Informationen finden Sie unter [Erfüllung der Compliance-Anforderungen mit S3 Replication Time Control \(S3RTC\)](#). API-Beispiele finden Sie [PutBucketReplication](#) in der Amazon Simple Storage Service API Reference.

Replizieren Sie Objekte, die mit serverseitiger Verschlüsselung erstellt wurden, mithilfe von AWS KMS

Ihr Quell-Bucket enthält möglicherweise Objekte, die mit serverseitiger Verschlüsselung erstellt wurden, indem Sie AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS). Standardmäßig repliziert Amazon S3 diese Objekte nicht. Optional können Sie Amazon S3 zum Replizieren dieser Objekte anweisen. Um dies zu tun, entscheiden Sie sich zunächst explizit für diese Funktion, indem Sie das `SourceSelectionCriteria`-Element hinzufügen. Geben Sie dann die AWS KMS key (für die AWS-Region des Ziel-Buckets), das für die Verschlüsselung von Objektreplikaten verwendet werden soll. Das folgende Beispiel zeigt, wie diese Elemente angegeben werden.

```
...
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
    <Status>Enabled</Status>
  </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
  <EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key ID to use for encrypting object replicas</
ReplicaKmsKeyID>
  </EncryptionConfiguration>
</Destination>
...
```

Weitere Informationen finden Sie unter [Replikation verschlüsselter Objekte \(SSE-C, SSE -S3, -, SSE -KMS\) DSSE KMS](#).

Beispiele für Replikations-Konfigurationen

Fügen Sie zunächst die folgenden Beispiel-Replikations-Konfigurationen zu Ihrem Bucket hinzu.

Important

Um eine Replikationskonfiguration zu einem Bucket hinzuzufügen, benötigen Sie die `iam:PassRole`-Berechtigung. Mit dieser Berechtigung können Sie die IAM Rolle weitergeben, die Amazon S3 S3-Replikationsberechtigungen gewährt. Sie geben die IAM Rolle an, indem Sie den Amazon-Ressourcennamen (ARN) angeben, der in dem `Role` Element in der Replikationskonfiguration verwendet wird XML. Weitere Informationen finden

Sie unter [Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an AWS-Service](#) im IAM-Benutzerhandbuch.

Example 1: Replikations-Konfiguration mit einer Regel

Die folgende grundlegende Replikations-Konfiguration legt eine Regel fest. Die Regel spezifiziert eine IAM Rolle, die Amazon S3 übernehmen kann, und einen einzelnen Ziel-Bucket für Objektreplikate. Der Status-Wert von Enabled gibt an, dass die Regel aktiviert ist.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>

    <Destination><Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Um eine Teilmenge von Objekten für die Replikation auszuwählen, können Sie einen Filter hinzufügen. In der folgenden Konfiguration gibt der Filter ein Objektschlüsselpräfix an. Diese Regel gilt für Objekte mit dem Schlüsselnamenpräfix *Tax/*.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
      <Prefix>Tax/</Prefix>
    </Filter>

    <Destination><Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket></Destination>

  </Rule>
```

```
</ReplicationConfiguration>
```

Wenn Sie das `Filter`-Element angeben, müssen Sie auch die Elemente `Priority` und `DeleteMarkerReplication` einschließen. In diesem Beispiel ist die `Priority` (Priorität) nicht relevant, da nur eine Regel vorhanden ist.

In der folgenden Konfiguration gibt der Filter ein Präfix und zwei Markierungen an. Die Regel gilt für eine Untermenge der Objekte, die das angegebene Schlüsselpräfix und die angegebenen Markierungen aufweisen. Insbesondere gilt es für ein Objekt, das das `Tax/`-Präfix in seinen Schlüsselnamen und den beiden angegebenen Objekt-Tags hat. Die `Priority` (Priorität) trifft nicht zu, weil es nur eine Regel gibt.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
      <And>
        <Prefix>Tax/</Prefix>
        <Tag>
          <Tag>
            <Key>tagA</Key>
            <Value>valueA</Value>
          </Tag>
        </Tag>
        <Tag>
          <Tag>
            <Key>tagB</Key>
            <Value>valueB</Value>
          </Tag>
        </Tag>
      </And>
    </Filter>

    <Destination><Bucket>arn:aws:s3::amzn-s3-demo-bucket</Bucket></Destination>
```

```
</Rule>
</ReplicationConfiguration>
```

Sie können eine Speicherklasse für die Objektreplikate wie folgt festlegen.

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
      <StorageClass>storage-class</StorageClass>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Sie können jede Speicherklasse festlegen, die Amazon S3 unterstützt.

Example 2: Replikations-Konfiguration mit zwei Regeln

Example

In der folgenden Replikations-Konfiguration:

- Jede Regel filtert nach einem anderen Schlüsselpräfix, sodass jede Regel auf eine bestimmte Teilmenge von Objekten angewendet wird. In diesem Beispiel repliziert Amazon S3 Objekte mit den Schlüsselnamen *Tax/doc1.pdf* and *Project/project1.txt*, aber es repliziert keine Objekte mit dem Schlüsselnamen *PersonalDoc/DocumentA*.
- Die Regelpriorität ist nicht relevant, da die Regeln für zwei unterschiedliche Mengen an Objekten gelten. Das nächste Beispiel zeigt, was geschieht, wenn die Regelpriorität angewendet wird.
- Die zweite Regel gibt eine Speicherklasse S3 Standard-IA für Objektreplikate an. Amazon S3 verwendet die angegebene Speicherklasse für diese Objektreplikate.

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
```

```

<Rule>
  <Status>Enabled</Status>
  <Priority>1</Priority>
  <DeleteMarkerReplication>
    <Status>string</Status>
  </DeleteMarkerReplication>
  <Filter>
    <Prefix>Tax</Prefix>
  </Filter>
  <Status>Enabled</Status>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
  </Destination>
  ...
</Rule>
<Rule>
  <Status>Enabled</Status>
  <Priority>2</Priority>
  <DeleteMarkerReplication>
    <Status>string</Status>
  </DeleteMarkerReplication>
  <Filter>
    <Prefix>Project</Prefix>
  </Filter>
  <Status>Enabled</Status>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    <StorageClass>STANDARD_IA</StorageClass>
  </Destination>
  ...
</Rule>

</ReplicationConfiguration>

```

Example 3: Replikations-Konfiguration mit zwei Regeln mit einander überlappenden Präfixen

In dieser Konfiguration spezifizieren die beiden Regeln Filter mit überlappenden Schlüsselpräfixen, **Stern/** und **Raumschiff/**Beide Regeln gelten für Objekte mit dem Schlüsselnamen **.Raumschiff-X**. In diesem Fall bestimmt Amazon S3 anhand der Regelpriorität, welche Regel angewendet werden soll. Je höher die Zahl, desto höher die Priorität.

```
<ReplicationConfiguration>
```

```

<Role>arn:aws:iam::account-id:role/role-name</Role>

<Rule>
  <Status>Enabled</Status>
  <Priority>1</Priority>
  <DeleteMarkerReplication>
    <Status>string</Status>
  </DeleteMarkerReplication>
  <Filter>
    <Prefix>star</Prefix>
  </Filter>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
  </Destination>
</Rule>
<Rule>
  <Status>Enabled</Status>
  <Priority>2</Priority>
  <DeleteMarkerReplication>
    <Status>string</Status>
  </DeleteMarkerReplication>
  <Filter>
    <Prefix>starship</Prefix>
  </Filter>
  <Destination>
    <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
  </Destination>
</Rule>
</ReplicationConfiguration>

```

Example 4. Beispielhafte Walkthroughs

Beispielanleitungen finden Sie unter [Beispiele für die Konfiguration der Live-Replikation](#).

Weitere Informationen zur XML Struktur der Replikationskonfiguration finden Sie [PutBucketReplication](#) in der Amazon Simple Storage Service API Reference.

Abwärtskompatibilität

Die neueste Version der Replikationskonfiguration XML ist V2. XMLV2-Replikationskonfigurationen enthalten das `Filter` Element für Regeln und Regeln, die S3 Replication Time Control (S3RTC) spezifizieren.

Um Ihre Version der Replikationskonfiguration zu sehen, können Sie den `GetBucketReplication` API Vorgang verwenden. Weitere Informationen finden Sie [GetBucketReplication](#) in der Amazon Simple Storage Service API Reference.

Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin die XML V1-Replikationskonfiguration. Wenn Sie die XML V1-Replikationskonfiguration verwendet haben, sollten Sie die folgenden Probleme berücksichtigen, die sich auf die Abwärtskompatibilität auswirken:

- Die Replikationskonfiguration XML V2 enthält das `Filter` Element für Regeln. Mit dem `Filter`-Element können Sie Objektfiler basierend auf dem Objektschlüsselpräfix, Tags oder einer Kombination aus beidem angeben, um die Objekte festzulegen, für die die Regel gilt. Die Replikationskonfiguration XML V1 unterstützt das Filtern nur auf der Grundlage des `key prefix`. In diesem Fall fügen Sie `Prefix` wie im folgenden Beispiel direkt als untergeordnetes Element des Elements `Rule` hinzu.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Prefix>key-prefix</Prefix>
    <Destination><Bucket>arn:aws:s3::amzn-s3-demo-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Zur Wahrung der Abwärtskompatibilität unterstützt Amazon S3 auch weiterhin die V1-Konfiguration.

- Wenn Sie ein Objekt aus Ihrem Quell-Bucket löschen, ohne eine Objektversions-ID anzugeben, fügt Amazon S3 eine Löschmarkierung hinzu. Wenn Sie Version 1 der Replikationskonfiguration verwenden XML, repliziert Amazon S3 Löschmarkierungen, die sich aus Benutzeraktionen ergeben. Anders ausgedrückt: Amazon S3 repliziert die Löschmarkierung nur, wenn ein Benutzer ein Objekt löscht. Wenn ein abgelaufenes Objekt von Amazon S3 (im Rahmen einer Lebenszyklusaktion) entfernt wird, repliziert Amazon S3 die Löschmarkierung nicht.

In V2-Replikationskonfigurationen können Sie die Replikation von Löschmarkierungen für non-tag-based Regeln aktivieren. Weitere Informationen finden Sie unter [Replizieren von Löschmarkierungen auf Buckets](#).

Berechtigungen für die Live-Replikation einrichten

Beim Einrichten der Live-Replikation müssen Sie die erforderlichen Berechtigungen wie folgt erwerben:

- Amazon S3 benötigt Berechtigungen, um Objekte in Ihrem Namen zu replizieren. Sie gewähren diese Berechtigungen, indem Sie eine IAM Rolle erstellen und diese Rolle dann in Ihrer Replikationskonfiguration angeben.
- Wenn sich der Quell- und der Ziel-Bucket im Besitz verschiedener Konten befinden, muss der Eigentümer des Ziel-Buckets dem Quell-Bucket-Eigentümer die Berechtigungen zum Speichern der Replikate erteilen.

Themen

- [Eine IAM Rolle erstellen](#)
- [Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten](#)
- [Gewähren von Berechtigungen für S3-Batch-Operationen](#)
- [Ändern des Replikatbesitzers](#)
- [Aktivieren des Empfangs replizierter Objekte aus einem Quell-Bucket](#)

Eine IAM Rolle erstellen

Standardmäßig sind alle Amazon-S3-Ressourcen – Buckets, Objekte und zugehörige Unterressourcen – privat, sodass nur der Ressourcenbesitzer auf die Ressource zugreifen kann. Amazon S3 benötigt Berechtigungen zum Lesen und Replizieren von Objekten aus dem Quell-Bucket. Sie gewähren diese Berechtigungen, indem Sie eine IAM Rolle erstellen und die Rolle in Ihrer Replikationskonfiguration angeben.

In diesem Abschnitt werden die Vertrauensrichtlinie und die mindestens erforderliche Berechtigungsrichtlinie erläutert. Die exemplarischen Vorgehensweisen enthalten step-by-step Anweisungen zum Erstellen einer IAM Rolle. Weitere Informationen finden Sie unter [Beispiele für die Konfiguration der Live-Replikation](#).

- Im folgenden Beispiel wird eine Vertrauensrichtlinie gezeigt, bei der Sie Amazon S3 als den Service-Prinzipal identifizieren, der die Rolle übernehmen kann.


```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Principal":{"
        "Service":"s3.amazonaws.com"
      },
      "Action":"sts:AssumeRole"
    }
  ]
}
```

- Im folgenden Beispiel wird eine Vertrauensrichtlinie gezeigt, bei der Sie Amazon S3 und S3-Batchvorgänge als Service-Prinzipale identifizieren. Dies ist nützlich, wenn Sie einen Batchreplikationsauftrag erstellen. Weitere Informationen finden Sie unter [Erstellen eines Batch-Replikationsauftrags für eine erste Replikationsregel oder ein neues Ziel](#).

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Principal":{"
        "Service": [
          "s3.amazonaws.com",
          "batchoperations.s3.amazonaws.com"
        ]
      },
      "Action":"sts:AssumeRole"
    }
  ]
}
```

Weitere Informationen zu IAM Rollen finden Sie im IAMBenutzerhandbuch unter [IAMRollen](#).

- Im folgenden Beispiel wird eine Zugriffsrichtlinie gezeigt, bei der Sie der Rolle die Berechtigungen erteilen, Replikationsaufgaben in Ihrem Namen durchzuführen. Wenn Amazon S3 die Rolle annimmt, verfügt es über die Berechtigungen, die Sie in dieser Richtlinie angeben. In dieser Richtlinie ist *amzn-s3-demo-bucket1* der Quell-Bucket und *amzn-s3-demo-bucket2* der Ziel-Bucket.

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:GetReplicationConfiguration",
        "s3:ListBucket"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-bucket1"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
      ],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket2/*"
    }
  ]
}


```

Die Zugriffsrichtlinie erteilt Berechtigungen für folgenden Aktionen:

- `s3:GetReplicationConfiguration` und `s3:ListBucket` – Berechtigungen für diese Aktionen im `amzn-s3-demo-bucket1`-Bucket (Quell-Bucket) erlauben es Amazon S3, die Replikationskonfiguration abzurufen und den Bucket-Inhalt aufzulisten. (Das aktuelle

Berechtigungsmodell erfordert die `s3:ListBucket`-Berechtigung für den Zugriff auf Löschmarkierungen.)

- `s3:GetObjectVersionForReplication` und `s3:GetObjectVersionAcl` — Berechtigungen für diese Aktionen werden für alle Objekte erteilt, sodass Amazon S3 eine bestimmte Objektversion und Zugriffskontrollliste (ACL) abrufen kann, die den Objekten zugeordnet sind.
- `s3:ReplicateObject` und `s3:ReplicateDelete` – Berechtigungen für diese Aktionen für alle Objekte im *amzn-s3-demo-bucket2*-Bucket (Ziel-Bucket) erlauben es Amazon S3, Objekte oder Löschmarkierungen in den Ziel-Bucket zu replizieren. Informationen zu Löschmarkierungen finden Sie unter [Auswirkungen von Löschvorgängen auf die Replikation](#).


 Note

Berechtigungen für die `s3:ReplicateObject` Aktion im *amzn-s3-demo-bucket2* Bucket (dem Ziel-Bucket) ermöglichen auch die Replikation von Metadaten wie Objekt-Tags und ACLs. Daher müssen Sie für die `s3:ReplicateTags`-Aktion keine explizite Berechtigung erteilen.

- `s3:GetObjectVersionTagging` – Berechtigungen für diese Aktion für Objekte im *amzn-s3-demo-bucket1*-Bucket (Quell-Bucket) gestatten es Amazon S3, Objekt-Tags für die Replikation zu lesen. Weitere Informationen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#). Wenn Amazon S3 nicht über diese Berechtigungen verfügt, repliziert es die Objekte, aber nicht die Objekt-Markierungen.

Eine Liste der Amazon S3-Aktionen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

 Important

Das Tool AWS-Konto dem die IAM Rolle gehört, muss über Berechtigungen für die Aktionen verfügen, die er der IAM Rolle gewährt.

Nehmen wir zum Beispiel an, dass der Quell-Bucket Objekte enthält, die einem anderen gehören AWS-Konto. Der Eigentümer der Objekte muss ausdrücklich die AWS-Konto der IAM Rolle die erforderlichen Berechtigungen über das Objekt besitztACL. Andernfalls

kann Amazon S3 nicht auf die Objekte zugreifen, und die Replikation dieser Objekte schlägt fehl. Hinweise zu ACL Berechtigungen finden Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Die hier beschriebenen Berechtigungen gehören zur Mindest-Replikationskonfiguration. Wenn Sie optionale Replikations-Konfigurationen hinzufügen möchten, müssen Sie Amazon S3 zusätzliche Berechtigungen erteilen.

Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören
AWS-Konten

Wenn sich der Quell- und der Ziel-Bucket im Besitz von unterschiedlichen Konten befinden, muss der Eigentümer des Ziel-Buckets auch eine Bucket-Richtlinie hinzufügen, um dem Eigentümer des Quell-Buckets die Berechtigung zum Ausführen von Replikationsaktionen wie folgt zu erteilen. In dieser Richtlinie ist *amzn-s3-demo-bucket2* der Ziel-Bucket.

Note

Das ARN Format der Rolle sieht möglicherweise anders aus. Wenn die Rolle mithilfe der Konsole erstellt wurde, lautet das ARN Format `arn:aws:iam::account-ID:role/service-role/role-name`. Wenn die Rolle mit dem erstellt wurde AWS CLI, das ARN Format ist `arn:aws:iam::account-ID:role/role-name`. Weitere Informationen finden Sie im IAM Benutzerhandbuch unter [IAM Rollen](#).

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForDestinationBucket",
  "Statement": [
    {
      "Sid": "Permissions on objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::SourceBucket-account-ID:role/service-role/source-account-IAM-role"
      },
      "Action": [
        "s3:ReplicateDelete",
        "s3:ReplicateObject"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket2/*"
  },
  {
    "Sid": "Permissions on bucket",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::SourceBucket-account-ID:role/service-role/source-account-IAM-role"
    },
    "Action": [
      "s3:List*",
      "s3:GetBucketVersioning",
      "s3:PutBucketVersioning"
    ],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket2"
  }
]
}

```

Ein Beispiel finden Sie unter [Konfigurieren der Replikation, wenn sich Quell- und Ziel-Buckets im Eigentum verschiedener Konten befinden](#).

Wenn Objekte im Quell-Bucket mit einem Tag versehen sind, beachten Sie Folgendes:

- Wenn der Besitzer des Quell-Buckets Amazon S3 die Erlaubnis für die `s3:GetObjectVersionTagging` und `s3:ReplicateTags` Aktionen zur Replikation von Objekt-Tags (über die IAM Rolle) erteilt, repliziert Amazon S3 die Tags zusammen mit den Objekten. Informationen zur IAM Rolle finden Sie unter [Eine IAM Rolle erstellen](#)
- Wenn der Eigentümer des Ziel-Buckets die Tags nicht replizieren will, kann er die folgende Anweisung zur Richtlinie für den Ziel-Bucket hinzufügen, um die Berechtigung für die Aktion `s3:ReplicateTags` explizit zu verweigern. In dieser Richtlinie `amzn-s3-demo-bucket2` ist der Ziel-Bucket.

```

...
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::SourceBucket-account-id:role/service-role/source-account-IAM-role"
      },
    },
  ],

```

```
    "Action": "s3:ReplicateTags",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket2/*"
  }
]
...
```

Gewähren von Berechtigungen für S3-Batch-Operationen

Die S3-Batch-Replikation bietet Ihnen eine Möglichkeit, Objekte zu replizieren, die existierten, bevor eine Replikationskonfiguration vorhanden war, Objekte, die zuvor repliziert wurden, und Objekte, bei denen die Replikation fehlgeschlagen ist. Sie können einen einmaligen Batch-Replikationsauftrag erstellen, wenn Sie die erste Regel in einer neuen Replikationskonfiguration erstellen oder wenn Sie einer vorhandenen Konfiguration ein neues Ziel hinzufügen, indem Sie AWS Management Console. Sie können die Batch-Replikation auch für eine bestehende Replikationskonfiguration initiieren, indem Sie einen Batch-Operations-Job erstellen.

IAM Rollen- und Richtlinienbeispiele für Batch-Replikation finden Sie unter [Konfiguration von IAM Richtlinien für die Batch-Replikation](#).

Ändern des Replikatbesitzers

Wenn unterschiedlich AWS-Konten wenn Sie die Quell- und Ziel-Buckets besitzen, können Sie Amazon S3 anweisen, den Eigentümer des Replikats auf den AWS-Konto dem der Ziel-Bucket gehört. Weitere Informationen zum Außerkraftsetzen des Besitzers finden Sie unter [Ändern des Replikat-Eigentümers](#).

Aktivieren des Empfangs replizierter Objekte aus einem Quell-Bucket

Sie können schnell die Richtlinien generieren, die für den Empfang replizierter Objekte aus einem Quell-Bucket erforderlich sind, und zwar über AWS Management Console.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket aus, den Sie als Ziel-Bucket verwenden möchten.
4. Wählen Sie die Registerkarte Management (Verwaltung) aus und scrollen Sie nach unten zu Replication rules (Replikationsregeln).
5. Wählen Sie für Actions (Aktionen) die Option Receive replicated objects (Replizierte Objekte empfangen) aus.

Folgen Sie den Anweisungen und geben Sie den AWS-Konto ID des Quell-Bucket-Kontos und wählen Sie Richtlinien generieren aus. Dadurch werden eine Amazon S3 S3-Bucket-Richtlinie und eine KMS Schlüsselrichtlinie generiert.

6. Wenn Sie diese Richtlinie Ihrer bestehenden Bucket-Richtlinie hinzufügen möchten, wählen Sie entweder Apply settings (Einstellungen anwenden) oder Copy (Kopieren) aus, um die Änderungen manuell zu kopieren.
7. (Optional) Kopieren Sie AWS KMS Richtlinie in die gewünschte KMS Schlüsselrichtlinie auf der AWS Key Management Service console.

Beispiele für die Konfiguration der Live-Replikation

Das folgende Beispiel zeigt das Konfigurieren der Live-Replikation für häufige Anwendungsfälle.

Note

Live-Replikation bezieht sich auf Replikation in derselben Region (SRR) und regionsübergreifende Replikation (). CRR Bei der Live-Replikation werden keine Objekte repliziert, die vor der Einrichtung der Replikation im Bucket vorhanden waren. Um Objekte zu replizieren, die vor der Einrichtung der Replikation vorhanden waren, verwenden Sie die On-Demand-Replikation. Informationen zum Synchronisieren von Buckets und zum Replizieren vorhandener Objekte bei Bedarf finden Sie unter. [Replizieren vorhandener Objekte](#)

Diese Beispiele zeigen, wie eine Replikationskonfiguration mithilfe der Amazon S3 S3-Konsole erstellt wird. AWS Command Line Interface (AWS CLI) und AWS SDKs (AWS SDK for Java and AWS SDK for .NET Beispiele werden gezeigt).

Informationen zur Installation und Konfiguration des AWS CLI, finden Sie zu den folgenden Themen in der AWS Command Line Interface Benutzerleitfaden.

- [Installation des AWS Command Line Interface](#)
- [Konfiguration des AWS CLI](#)— Sie müssen mindestens ein Profil einrichten. Richten Sie bei kontoübergreifenden Szenarien zwei Profile ein.

Für Informationen über AWS SDKs, siehe [AWS SDK für Java](#) und [AWS SDK für .NET](#).

i Tip

Ein step-by-step Tutorial, das zeigt, wie Sie Daten mithilfe der Live-Replikation replizieren können, finden Sie unter [Praktische Einführung: Daten innerhalb und zwischen Daten replizieren AWS-Regionen mithilfe der S3-Replikation](#).

Themen

- [Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos](#)
- [Konfigurieren der Replikation, wenn sich Quell- und Ziel-Buckets im Eigentum verschiedener Konten befinden](#)
- [Erfüllung der Compliance-Anforderungen mit S3 Replication Time Control \(S3RTC\)](#)
- [Replikation verschlüsselter Objekte \(SSE-C, SSE -S3, -, SSE -KMS\) DSSE KMS](#)
- [Replizieren von Metadatenänderungen mit der Synchronisierung von Amazon-S3-Replikatänderungen](#)
- [Replizieren von Löschmarkierungen auf Buckets](#)

Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos

Replikation ist das automatische, asynchrone Kopieren von Objekten über Buckets hinweg, die sich im selben oder in unterschiedlichen Buckets befinden AWS-Regionen. Bei der Replikation werden neu erstellte Objekte und Objektaktualisierungen von einem Quell-Bucket in einen oder mehrere Ziel-Buckets kopiert. Weitere Informationen finden Sie unter [Überblick über das Replizieren von Objekten](#).

Wenn Sie die Replikation konfigurieren, fügen Sie dem Quell-Bucket Replikationsregeln hinzu. Replikationsregeln definieren, welche Quell-Bucket-Objekte repliziert werden sollen, und den Ziel-Bucket/die Ziel-Buckets, in dem/denen die replizierten Objekte gespeichert werden sollen. Sie können eine Regel erstellen, um alle Objekte in einem Bucket oder eine Untermenge von Objekten mit einem spezifischen Schlüsselnamenpräfixen, einem oder mehreren Objekt-Markierungen oder beidem zu replizieren. Ein Ziel-Bucket kann sich in demselben befinden AWS-Konto wie der Quell-Bucket, oder er kann sich in einem anderen Konto befinden.

Wenn Sie angeben, dass eine Objektversions-ID gelöscht werden soll, löscht Amazon S3 diese Objektversion im Quell-Bucket. Es repliziert die Löschung aber nicht im Ziel-Bucket. Anders ausgedrückt: Dieselbe Objektversion wird im Ziel-Bucket nicht gelöscht. Dies schützt Daten vor missbräuchlichen Löschungen.

Wenn Sie einem Bucket eine Replikationsregel hinzufügen, ist diese standardmäßig aktiviert, sodass sie ausgeführt wird, sobald Sie sie speichern.

In diesem Beispiel richten Sie die Replikation für Quell- und Ziel-Buckets ein, die demselben Unternehmen gehören AWS-Konto. Es werden Beispiele für die Verwendung der Amazon S3 S3-Konsole bereitgestellt, der AWS Command Line Interface (AWS CLI) und die AWS SDK for Java and AWS SDK for .NET.


Verwenden der S3-Konsole

Um eine Replikationsregel zu konfigurieren, wenn sich der Ziel-Bucket im selben befindet AWS-Konto. Gehen Sie wie beim Quell-Bucket wie folgt vor.

Wenn sich der Ziel-Bucket in einem anderen Konto als der Quell-Bucket befindet, müssen Sie eine Bucket-Richtlinie für den Ziel-Bucket hinzufügen, um dem Eigentümer des Quell-Bucket-Kontos die Berechtigung zum Replizieren von Objekten in der Ziel-Bucket zu erteilen. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten](#).

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
4. Wählen Sie die Registerkarte Verwaltung aus, scrollen Sie nach unten zu Replikationsregeln und wählen Sie dann Replikationsregel erstellen aus.
5. Geben Sie im Abschnitt Replikationsregelkonfiguration unter Name der Replikationsregel einen Namen für Ihre Regel ein, um die Regel später leichter identifizieren zu können. Der Name ist erforderlich und muss innerhalb des Buckets eindeutig sein.
6. In Status (Status) ist Enabled (Aktiviert) standardmäßig ausgewählt. Eine aktivierte Regel wird ausgeführt, sobald Sie speichern. Wenn Sie die Regel später aktivieren möchten, wählen Sie Deaktiviert aus.
7. Wenn es bereits Replikationsregeln für den Bucket gibt, werden Sie angewiesen, eine Priorität für die Regel festzulegen. Sie müssen eine Priorität für die Regel festlegen, um Konflikte zu vermeiden, die durch Objekte verursacht werden, die mehreren Regeln unterliegen. Wenn sich Regeln überschneiden, verwendet Amazon S3 die Regelpriorität, um die Regel zu ermitteln, die angewendet werden muss. Je höher die Zahl, desto höher die Priorität. Weitere Informationen zur Priorität von Regeln finden Sie unter [Replikationskonfiguration](#).

8. Unter Quell-Bucket stehen Ihnen folgende Optionen zum Festlegen der Replikationsquelle zur Verfügung:
- Um den gesamten Bucket zu replizieren, wählen Sie Apply to all objects in the bucket (Auf alle Objekte im Bucket anwenden).
 - Um alle Objekte zu replizieren, die dasselbe Präfix haben, wählen Sie Limit the scope of this rule using one or more filters (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken). Dies beschränkt die Replikation auf alle Objekte, deren Namen mit dem von Ihnen angegebenen Präfix beginnen (z. B. pictures). Geben Sie ein Präfix in das Feld Präfix ein.


 Note

Wenn Sie ein Präfix eingeben, das den Namen eines Ordners darstellt, müssen Sie / (Schrägstrich) als letztes Zeichen eingeben (z. B. pictures/).

- Um alle Objekte mit einem oder mehreren Objekt-Tags zu replizieren, wählen Sie Tag hinzufügen aus und geben Sie das Schlüssel-Wert-Paar in die Felder ein. Wiederholen Sie den Vorgang, um ein weiteres Tag hinzuzufügen. Sie können ein Präfix und Markierungen kombinieren. Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

Das neue XML Replikationskonfigurationsschema unterstützt Präfix- und Tag-Filterung sowie die Priorisierung von Regeln. Weitere Informationen über das neue Schema finden Sie unter [Abwärtskompatibilität](#). Weitere Informationen zur XML Verwendung mit Amazon S3API, die hinter der Benutzeroberfläche steckt, finden Sie unter [Replikationskonfiguration](#). Das neue Schema wird als Replikationskonfiguration XML V2 beschrieben.

9. Wählen Sie unter Ziel den Bucket aus, in den Amazon S3 Objekte replizieren soll.

 Note

Die Anzahl der Ziel-Buckets ist begrenzt auf die Anzahl von AWS-Regionen in einer bestimmten Partition. Eine Partition ist eine Gruppe von Regionen. AWS hat derzeit drei Partitionen: aws (Standardregionen), aws-cn (China-Regionen) und aws-us-gov (AWS GovCloud (US) Regionen). Sie können [Service Quotas](#) verwenden, um eine Erhöhung Ihres Ziel-Bucket-Kontingents anzufordern.

- Um in einen Bucket oder mehrere Buckets in Ihrem Konto zu replizieren, wählen Sie Bucket in diesem Konto wählen aus und geben Sie die Ziel-Bucket-Namen ein oder suchen Sie danach.
- Um in einen Bucket oder Buckets in einem anderen zu replizieren AWS-Konto, wählen Sie Spezifizieren Sie einen Bucket in einem anderen Konto und geben Sie die Konto-ID und den Bucket-Namen des Ziel-Buckets ein.

Wenn sich das Ziel in einem anderen Konto als der Quell-Bucket befindet, müssen Sie eine Bucket-Richtlinie für die Ziel-Buckets hinzufügen, um dem Eigentümer des Quell-Bucket-Kontos die Berechtigung zum Replizieren von Objekten zu erteilen. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten](#).

Wenn Sie eine leichtere Standardisierung der Eigentümerschaft für neue Objekte im Ziel-Bucket ermöglichen möchten, wählen Sie Objekteigentümerschaft in Eigentümer des Ziel-Buckets ändern aus. Weitere Informationen zu dieser Option finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Note

Wenn die Versionssteuerung für den Ziel-Bucket nicht aktiviert ist, erhalten Sie eine Warnmeldung, die die Schaltfläche Versionierung aktivieren enthält. Wählen Sie diese Schaltfläche, um das Versioning für den Bucket zu aktivieren.

10. Richten Sie ein AWS Identity and Access Management (IAM) Rolle, die Amazon S3 übernehmen kann, um Objekte in Ihrem Namen zu replizieren.

Um eine IAM Rolle einzurichten, wählen Sie im IAMRollenbereich eine der folgenden Optionen aus der IAMRollen-Dropdownliste aus:

- Wir empfehlen dringend, dass Sie Neue Rolle erstellen wählen, damit Amazon S3 eine neue IAM Rolle für Sie erstellt. Wenn Sie die Regel speichern, wird eine neue Richtlinie für die IAM Rolle generiert, die den von Ihnen ausgewählten Quell- und Ziel-Buckets entspricht.
- Sie können wählen, ob Sie eine vorhandene IAM Rolle verwenden möchten. In diesem Fall müssen Sie eine Rolle auswählen, die Amazon S3 die erforderlichen Berechtigungen für die Replikation gewährt. Die Replikation schlägt fehl, wenn diese Rolle Amazon S3 keine ausreichenden Berechtigungen gewährt, um Ihre Replikationsregel zu befolgen.

⚠ Important

Wenn Sie einem Bucket eine Replikationsregel hinzufügen, benötigen Sie die `iam:PassRole` Erlaubnis, die IAM Rolle zu übergeben, die Amazon S3 S3-Replikationsberechtigungen gewährt. Weitere Informationen finden Sie unter [Einem Benutzer Berechtigungen zur Übergabe einer Rolle an einen AWS-Service](#) im IAM-Benutzerhandbuch.

11. Um Objekte im Quell-Bucket zu replizieren, die mit serverseitiger Verschlüsselung verschlüsselt sind, mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS), wählen Sie unter Verschlüsselung die Option Objekte replizieren, die verschlüsselt sind mit AWS KMS. Unter AWS KMS Schlüssel zum Verschlüsseln von Zielobjekten sind die Quellschlüssel, deren Verwendung Sie bei der Replikation zulassen. Alle KMS Quellschlüssel sind standardmäßig enthalten. Um die KMS Schlüsselauswahl einzuschränken, können Sie einen Alias oder eine Schlüssel-ID wählen.

Objekte verschlüsselt von AWS KMS keys die Sie nicht auswählen, werden nicht repliziert. Ein KMS Schlüssel oder eine Gruppe von KMS Schlüsseln wird für Sie ausgewählt, aber Sie können die KMS Schlüssel wählen, wenn Sie möchten. Für Informationen zur Verwendung AWS KMS mit Replikation finden Sie unter [Replikation verschlüsselter Objekte \(SSE-C, SSE -S3, -, SSE - KMS\) DSSE KMS](#).

⚠ Important

Wenn Sie Objekte replizieren, die verschlüsselt sind mit AWS KMS, der AWS KMS Die Anforderungsrate verdoppelt sich in der Quellregion und steigt in der Zielregion um den gleichen Betrag. Dadurch erhöhten sich die Anrufraten auf AWS KMS sind auf die Art und Weise zurückzuführen, wie Daten mithilfe des KMS Schlüssels, den Sie für die Zielregion der Replikation definieren, erneut verschlüsselt werden. AWS KMS hat ein Kontingent für die Anforderungsrate, das pro Anrufraten und Region gilt. Informationen zu den Standardkontingenten finden Sie unter [AWS KMS Kontingente — Anfragen pro Sekunde: Variiert](#) in AWS Key Management Service Leitfaden für Entwickler.

Wenn Ihre aktuelle Amazon S3 PUT S3-Objektanforderungsrate während der Replikation mehr als die Hälfte der Standardrate beträgt AWS KMS Wir empfehlen Ihnen, eine Erhöhung Ihrer Rate für Ihr Konto zu beantragen AWS KMS Preiskontingent anfragen. Um eine Erhöhung zu beantragen, erstellen Sie einen Fall im AWS Support Center

unter [Kontaktieren Sie uns](#). Nehmen wir zum Beispiel an, dass Ihre aktuelle PUT Objektanforderungsrate 1.000 Anfragen pro Sekunde beträgt und Sie AWS KMS um Ihre Objekte zu verschlüsseln. In diesem Fall empfehlen wir Ihnen, nachzufragen AWS Support um deine zu erhöhen AWS KMS Ratenlimit auf 2.500 Anfragen pro Sekunde, sowohl in Ihrer Quell- als auch in Ihrer Zielregion (falls unterschiedlich), um sicherzustellen, dass keine Drosselung durch AWS KMS.

Ihre PUT Objektanforderungsrate im Quell-Bucket finden Sie PutRequests in den CloudWatch Amazon-Anforderungsmetriken für Amazon S3. Informationen zum Anzeigen von CloudWatch Metriken finden Sie unter [Verwenden der S3-Konsole](#).

Wenn Sie Objekte replizieren möchten, die verschlüsselt sind mit AWS KMS wie folgt:

- Unter AWS KMS key Um Zielobjekte zu verschlüsseln, geben Sie Ihren KMS Schlüssel auf eine der folgenden Arten an:
 - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys, und wählen Sie Ihren KMS Schlüssel aus der Liste der verfügbaren Schlüssel aus.

Sowohl die Von AWS verwalteter Schlüssel (aws/s3) und Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter [Kundenschlüssel und AWS Schlüssel](#) im AWS Key Management Service Leitfaden für Entwickler.

- Um den KMS Schlüssel Amazon Resource Name (ARN) einzugeben, wählen Sie Enter AWS KMS key ARN, und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein. Dadurch werden die Replikate im Ziel-Bucket verschlüsselt. Sie finden den ARN für Ihren KMS Schlüssel in der [IAMKonsole](#) unter Verschlüsselungsschlüssel.
- Um einen neuen, vom Kunden verwalteten Schlüssel zu erstellen, klicken Sie auf AWS KMS Konsole, wählen Sie Create a KMS Key aus.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler.

⚠ Important

Sie können nur KMS Schlüssel verwenden, die in demselben aktiviert sind AWS-Region wie der Eimer. Wenn Sie „Aus Ihren KMS Schlüsseln auswählen“ wählen, listet die S3-Konsole nur 100 KMS Schlüssel pro Region auf. Wenn Sie mehr als 100 KMS Schlüssel in derselben Region haben, können Sie nur die ersten 100 KMS Schlüssel in der S3-Konsole sehen. Um einen KMS Schlüssel zu verwenden, der nicht in der Konsole aufgeführt ist, wählen Sie `Enter AWS KMS key ARN`, und geben Sie Ihren KMS Schlüssel einARN.


Wenn Sie eine verwenden AWS KMS key Für die serverseitige Verschlüsselung in Amazon S3 müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter [Identifizieren symmetrischer und KMS asymmetrischer Schlüssel](#) in der AWS Key Management Service Leitfaden für Entwickler.

Weitere Informationen zum Erstellen eines AWS KMS key, siehe [Schlüssel erstellen](#) in der AWS Key Management Service Leitfaden für Entwickler. Weitere Informationen zur Verwendung AWS KMS mit Amazon S3, siehe [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).

12. Wenn Sie Ihre Daten in eine bestimmte Speicherklasse im Ziel replizieren wollen, wählen Sie unter Zielspeicherklasse die Option Speicherklasse für die replizierten Objekte ändern aus. Anschließend wählen Sie die Speicherklasse, die Sie für die replizierten Objekte im Ziel verwenden wollen. Wenn Sie diese Option nicht auswählen, ist die Speicherklasse für replizierte Objekte dieselbe Klasse wie für die ursprünglichen Objekte.
13. Beim Festlegen der zusätzlichen Replikationsoptionen haben Sie folgende zusätzliche Optionen:
 - Wenn Sie S3 Replication Time Control (S3RTC) in Ihrer Replikationskonfiguration aktivieren möchten, wählen Sie Replication Time Control (RTC). Weitere Informationen zu dieser Option finden Sie unter [Erfüllung der Compliance-Anforderungen mit S3 Replication Time Control \(S3RTC\)](#).
 - Wenn Sie S3-Replikationsmetriken in Ihrer Replikationskonfiguration aktivieren möchten, wählen Sie Replication metrics and events (Replikationsmetriken und -ereignisse) aus.

Weitere Informationen finden Sie unter [Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status](#).

- Wenn Sie die Löschkennzeichnungs-Replikation in Ihrer Replikations-Konfiguration aktivieren möchten, wählen Sie Kennzeichnungsreplikation löschen aus. Weitere Informationen finden Sie unter [Replizieren von Löschkennzeichnungen auf Buckets](#).
- Wenn Sie die Synchronisierung der Amazon-S3-Replikatänderung in Ihrer Replikations-Konfiguration aktivieren möchten, wählen Sie Synchronisierung der Replikatänderung aus. Weitere Informationen finden Sie unter [Replizieren von Metadatenänderungen mit der Synchronisierung von Amazon-S3-Replikatänderungen](#).

 Note

Wenn Sie S3 RTC - oder S3-Replikationsmetriken verwenden, fallen zusätzliche Gebühren an.

14. Zum Abschluss wählen Sie Save (Speichern).
15. Nachdem Sie Ihre Regel gespeichert haben, können Sie Ihre Regel bearbeiten, aktivieren, deaktivieren oder löschen, indem Sie Ihre Regel auswählen und Edit rule (Regel bearbeiten) wählen.

Verwendung der AWS CLI

Um das AWS CLI um die Replikation einzurichten, wenn Quell- und Ziel-Buckets demselben gehören AWS-Konto, gehen Sie wie folgt vor:

- Erstellen Sie Quell- und Ziel-Buckets
- Aktivieren Sie das Versioning für die Buckets
- Erstellen Sie eine IAM Rolle, die Amazon S3 die Berechtigung erteilt, Objekte zu replizieren
- Fügen Sie die Replikationskonfiguration zum Quell-Bucket hinzu

Um die Einrichtung zu prüfen, testen Sie sie.

Um die Replikation einzurichten, wenn Quell- und Ziel-Buckets demselben gehören AWS-Konto

1. Richten Sie ein Anmeldeinformationsprofil ein für AWS CLI. In diesem Beispiel verwenden wir den Profilnamen `acctA`. Informationen zum Einrichten von Profilen mit Anmeldeinformationen finden Sie unter [Benannte Profile](#) in der AWS Command Line Interface Benutzerhandbuch.

 **Important**


Das Profil, das Sie für diese Übung verwenden, muss über die nötigen Berechtigungen verfügen. In der Replikationskonfiguration geben Sie beispielsweise die IAM Rolle an, die Amazon S3 übernehmen kann. Dies können Sie nur tun, wenn das verwendete Profil über die `iam:PassRole`-Berechtigung verfügt. Weitere Informationen finden Sie unter [Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an einen AWS Service](#) im IAM Benutzerhandbuch. Wenn Sie zur Erstellung eines benannten Profils die Anmeldeinformationen eines Administrators verwenden, können Sie alle Aufgaben durchführen.

2. Erstellen Sie einen *source*-Bucket und aktivieren Sie dafür das Versioning. Der folgende Code erstellt einen *source*-Bucket in der Region USA Ost (Nord-Virginia) (`us-east-1`).

```
aws s3api create-bucket \  
--bucket source \  
--region us-east-1 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket source \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

3. Erstellen Sie einen *destination*-Bucket und aktivieren Sie dafür das Versioning. Der folgende Code erstellt einen *destination*-Bucket in der Region US West (Oregon) (`us-west-2`).

 **Note**

Um die Replikationskonfiguration einzurichten, wenn sich Quell- und Ziel-Bucket im selben Bereich befinden AWS-Konto, verwenden Sie dasselbe Profil. Dieses Beispiel verwendet `acctA`. Um die Replikationskonfiguration zu testen, wenn die

Buckets verschiedenen Besitzern gehören AWS-Konten, geben Sie für jedes Profil unterschiedliche Profile an. In diesem Beispiel verwenden wir das Profil `acctB` für den Ziel-Bucket.

```
aws s3api create-bucket \  
--bucket destination \  
--region us-west-2 \  
--create-bucket-configuration LocationConstraint=us-west-2 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket destination \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

4. Erstellen Sie eine IAM Rolle. Sie geben diese Rolle in der Replikationskonfiguration an, die Sie der *source* Bucket später. Amazon S3 übernimmt diese Rolle, um Objekte in Ihrem Namen zu replizieren. Sie erstellen eine IAM Rolle in zwei Schritten:

- Erstellen Sie eine Rolle.
- Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

a. Erstellen Sie die IAM Rolle.

- i. Kopieren Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-trust-policy.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie gewährt Amazon S3 Service-Prinzipal-Berechtigungen, um die Rolle anzunehmen.

```
{  
  "Version":"2012-10-17",  
  "Statement":[  
    {  
      "Effect":"Allow",  
      "Principal":{  
        "Service":"s3.amazonaws.com"  
      },  
    ],  
}
```

```
        "Action": "sts:AssumeRole"
    }
  ]
}
```

- ii. Führen Sie den folgenden Befehl aus, um eine Rolle zu erstellen.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json \
--profile acctA
```

- b. Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

- i. Kopieren Sie die folgende Berechtigungsrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-permissions-policy.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Zugriffsrichtlinie erteilt Berechtigungen für verschiedene Amazon-S3-Bucket- und -Objektaktionen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::source-bucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::source-bucket"
      ]
    }
  ],
}
```

```

    {
      "Effect": "Allow",
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
      ],
      "Resource": "arn:aws:s3:::destination-bucket/*"
    }
  ]
}

```

- ii. Führen Sie den folgenden Befehl aus, um eine Richtlinie zu erstellen und sie der Rolle anzufügen.

```

$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file:///s3-role-permissions-policy.json \
--policy-name replicationRolePolicy \
--profile acctA

```

5. Fügen Sie eine Replikationskonfiguration zum *source*-Bucket hinzu.

- a. Amazon S3 API erfordert zwar eine Replikationskonfiguration als XML AWS CLI erfordert, dass Sie die Replikationskonfiguration als angebenJSON. Speichern Sie Folgendes JSON in einer Datei namens `replication.json` im lokalen Verzeichnis auf Ihrem Computer.

```

{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket": "arn:aws:s3:::destination-bucket"
      }
    }
  ]
}

```


- b. Aktualisieren Sie das, JSON indem Sie Werte für *destination-bucket* und angeben *IAM-role-ARN*. Speichern Sie die Änderungen.
- c. Führen Sie den folgenden Befehl aus, um die Replikations-Konfiguration zu Ihrem Quell-Bucket hinzuzufügen. Stellen Sie sicher, dass Sie den Namen für den *source*-Bucket angeben.

```
$ aws s3api put-bucket-replication \  
--replication-configuration file://replication.json \  
--bucket source \  
--profile acctA
```

Um die Replikationskonfiguration abzurufen, verwenden Sie den Befehl `get-bucket-replication`.

```
$ aws s3api get-bucket-replication \  
--bucket source \  
--profile acctA
```

6. Testen Sie das Setup in der Amazon-S3-Konsole:
 - a. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
 - b. Erstellen Sie im *source*-Bucket einen Ordner mit dem Namen Tax.
 - c. Fügen Sie Beispielobjekte zum Tax-Ordner im *source*-Bucket hinzu.

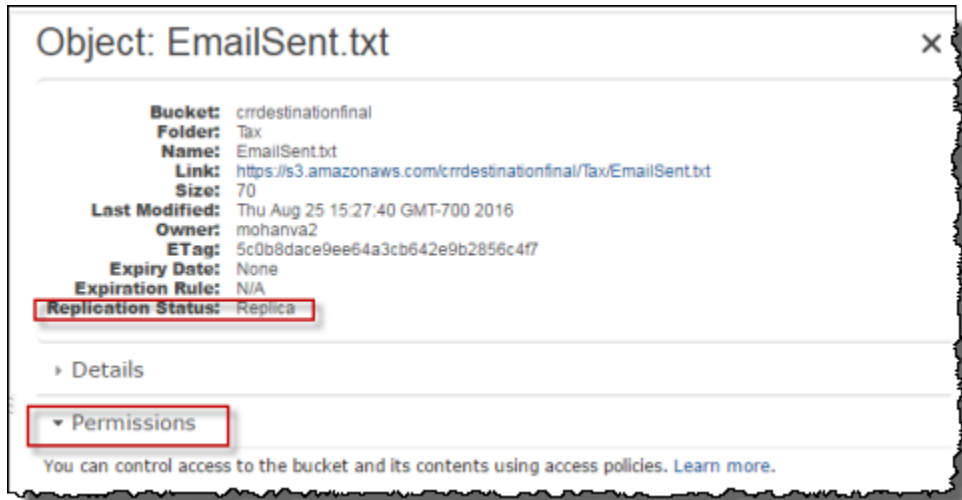
 Note

Die von Amazon S3 für die Replikation eines Objekts benötigte Zeit hängt von der Größe des Objekts ab. Weitere Informationen zum Anzeigen des Replikationsstatus finden Sie unter [Abrufen von Replikationsstatusinformationen](#).

Überprüfen Sie im *destination*-Bucket Folgendes:

- Dass Amazon S3 die Objekte repliziert hat.
- In den properties (Eigenschaften) des Objekts, dass der Replication Status (Replikationsstatus) auf `Replica` festgelegt ist (was es als Replikatobjekt kennzeichnet).

- In den properties (Eigenschaften) des Objekts, dass im Berechtigungsabschnitt keine Berechtigungen aufgeführt sind. Dies bedeutet, dass sich das Replikat noch im Besitz des *source*-Bucket-Eigentümers befindet und der *destination*-Bucket-Eigentümer über keine Berechtigung auf dem Objektreplikat verfügt. Sie können eine optionale Konfiguration hinzufügen, um Amazon S3 anzuweisen, die Replikat-Eigentümerschaft zu ändern. Ein Beispiel finden Sie unter [Wie ändert man den Besitzer des Replikats](#).



Verwendung der AWS SDKs

Verwenden Sie die folgenden Codebeispiele, um eine Replikationskonfiguration zu einem Bucket mit dem hinzuzufügen AWS SDK for Java and AWS SDK for .NET jeweils.

Java

Das folgende Beispiel fügt eine Replikations-Konfiguration einem Bucket hinzu und ruft die Konfiguration anschließend ab und überprüft sie. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.identitymanagement.AmazonIdentityManagement;
```

```
import
    com.amazonaws.services.identitymanagement.AmazonIdentityManagementClientBuilder;
import com.amazonaws.services.identitymanagement.model.CreateRoleRequest;
import com.amazonaws.services.identitymanagement.model.PutRolePolicyRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.BucketReplicationConfiguration;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.DeleteMarkerReplication;
import com.amazonaws.services.s3.model.DeleteMarkerReplicationStatus;
import com.amazonaws.services.s3.model.ReplicationDestinationConfig;
import com.amazonaws.services.s3.model.ReplicationRule;
import com.amazonaws.services.s3.model.ReplicationRuleStatus;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.replication.ReplicationFilter;
import com.amazonaws.services.s3.model.replication.ReplicationFilterPredicate;
import com.amazonaws.services.s3.model.replication.ReplicationPrefixPredicate;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class CrossRegionReplication {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String accountId = "**** Account ID ****";
        String roleName = "**** Role name ****";
        String sourceBucketName = "**** Source bucket name ****";
        String destBucketName = "**** Destination bucket name ****";
        String prefix = "Tax/";

        String roleARN = String.format("arn:aws:iam::%s:%s", accountId,
roleName);

        String destinationBucketARN = "arn:aws:s3:::" + destBucketName;

        AmazonS3 s3Client = AmazonS3Client.builder()
            .withCredentials(new ProfileCredentialsProvider())
            .withRegion(clientRegion)
            .build();
```

```
        createBucket(s3Client, clientRegion, sourceBucketName);
        createBucket(s3Client, clientRegion, destBucketName);
        assignRole(roleName, clientRegion, sourceBucketName,
destBucketName);

        try {

            // Create the replication rule.
            List<ReplicationFilterPredicate> andOperands = new
ArrayList<ReplicationFilterPredicate>();
            andOperands.add(new ReplicationPrefixPredicate(prefix));

            Map<String, ReplicationRule> replicationRules = new
HashMap<String, ReplicationRule>();
            replicationRules.put("ReplicationRule1",
                new ReplicationRule()
                    .withPriority(0)

.withStatus(ReplicationRuleStatus.Enabled)

.withDeleteMarkerReplication(
                                                                    new
DeleteMarkerReplication().withStatus(
                DeleteMarkerReplicationStatus.DISABLED))
                                                                    .withFilter(new
ReplicationFilter().withPredicate(
                                                                    new
ReplicationPrefixPredicate(prefix)))
                                                                    .withDestinationConfig(new
ReplicationDestinationConfig()

.withBucketARN(destinationBucketARN)

.withStorageClass(StorageClass.Standard)));

            // Save the replication rule to the source bucket.
            s3Client.setBucketReplicationConfiguration(sourceBucketName,
                new BucketReplicationConfiguration()
                    .withRoleARN(roleARN)

.withRules(replicationRules));
```

```
        // Retrieve the replication configuration and verify that
the configuration
        // matches the rule we just set.
        BucketReplicationConfiguration replicationConfig = s3Client
.getBucketReplicationConfiguration(sourceBucketName);
        ReplicationRule rule =
replicationConfig.getRule("ReplicationRule1");
        System.out.println("Retrieved destination bucket ARN: "
            +
rule.getDestinationConfig().getBucketARN());
        System.out.println("Retrieved priority: " +
rule.getPriority());
        System.out.println("Retrieved source-bucket replication rule
status: " + rule.getStatus());
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3
couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the
client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}

private static void createBucket(AmazonS3 s3Client, Regions region, String
bucketName) {
    CreateBucketRequest request = new CreateBucketRequest(bucketName,
region.getName());
    s3Client.createBucket(request);
    BucketVersioningConfiguration configuration = new
BucketVersioningConfiguration()
        .withStatus(BucketVersioningConfiguration.ENABLED);

    SetBucketVersioningConfigurationRequest enableVersioningRequest =
new SetBucketVersioningConfigurationRequest(
        bucketName, configuration);
    s3Client.setBucketVersioningConfiguration(enableVersioningRequest);
}
```



```

        private static void assignRole(String roleName, Regions region, String
sourceBucket, String destinationBucket) {
            AmazonIdentityManagement iamClient =
AmazonIdentityManagementClientBuilder.standard()
                .withRegion(region)
                .withCredentials(new ProfileCredentialsProvider())
                .build();

            StringBuilder trustPolicy = new StringBuilder();
            trustPolicy.append("{\r\n  ");
            trustPolicy.append("\\\\"Version\\":\\"2012-10-17\\",\r\n  ");
            trustPolicy.append("\\\\"Statement\\":[\r\n      {\r\n
");
                trustPolicy.append("\\\\"Effect\\":\\"Allow\\",\r\n      \\\
\\"Principal\\":{\r\n          ");
                trustPolicy.append("\\\\"Service\\":\\"s3.amazonaws.com\\",\r\n
            },\r\n          ");
                trustPolicy.append("\\\\"Action\\":\\"sts:AssumeRole\\",\r\n
            ]\r\n      ]\r\n}");

            CreateRoleRequest createRoleRequest = new CreateRoleRequest()
                .withRoleName(roleName)

.withAssumeRolePolicyDocument(trustPolicy.toString());

            iamClient.createRole(createRoleRequest);

            StringBuilder permissionPolicy = new StringBuilder();
            permissionPolicy.append(
                "{\r\n  \\\\"Version\\":\\"2012-10-17\\",\r\n
            \\\\"Statement\\":[\r\n      {\r\n          ");
            permissionPolicy.append(
                "\\\\"Effect\\":\\"Allow\\",\r\n          \\\
\\"Action\\":[\r\n          ");
            permissionPolicy.append("\\\\"s3:GetObjectVersionForReplication\\",\r\n
            ");
            permissionPolicy.append(
                "\\\\"s3:GetObjectVersionAcl\\",\r\n          ],\r\n
            \\\\"Resource\\":[\r\n          ");
            permissionPolicy.append("\\\\"arn:aws:s3::");
            permissionPolicy.append(sourceBucket);
            permissionPolicy.append("/.*\\",\r\n          ]\r\n          },\r\n
            {\r\n          ");
            permissionPolicy.append(

```



```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CrossRegionReplicationTest
    {
        private const string sourceBucket = "**** source bucket ****";
        // Bucket ARN example - arn:aws:s3:::destinationbucket
        private const string destinationBucketArn = "**** destination bucket ARN
****";
        private const string roleArn = "**** IAM Role ARN ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint sourceBucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(sourceBucketRegion);
            EnableReplicationAsync().Wait();
        }
        static async Task EnableReplicationAsync()
        {
            try
            {
                ReplicationConfiguration replConfig = new ReplicationConfiguration
                {
                    Role = roleArn,
                    Rules =
                    {
                        new ReplicationRule
                        {
                            Prefix = "Tax",
                            Status = ReplicationRuleStatus.Enabled,
                            Destination = new ReplicationDestination
                            {
                                BucketArn = destinationBucketArn
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
        };

        PutBucketReplicationRequest putRequest = new
PutBucketReplicationRequest
        {
            BucketName = sourceBucket,
            Configuration = replConfig
        };

        PutBucketReplicationResponse putResponse = await
s3Client.PutBucketReplicationAsync(putRequest);

        // Verify configuration by retrieving it.
        await RetrieveReplicationConfigurationAsync(s3Client);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}
private static async Task RetrieveReplicationConfigurationAsync(IAmazonS3
client)
{
    // Retrieve the configuration.
    GetBucketReplicationRequest getRequest = new GetBucketReplicationRequest
    {
        BucketName = sourceBucket
    };
    GetBucketReplicationResponse getResponse = await
client.GetBucketReplicationAsync(getRequest);
    // Print.
    Console.WriteLine("Printing replication configuration information...");
    Console.WriteLine("Role ARN: {0}", getResponse.Configuration.Role);
    foreach (var rule in getResponse.Configuration.Rules)
    {
        Console.WriteLine("ID: {0}", rule.Id);
        Console.WriteLine("Prefix: {0}", rule.Prefix);
        Console.WriteLine("Status: {0}", rule.Status);
    }
}
```

```
}  
  }  
} }
```

Konfigurieren der Replikation, wenn sich Quell- und Ziel-Buckets im Eigentum verschiedener Konten befinden

Einrichtung der Replikation wenn *source* and *destination* Buckets gehören verschiedenen AWS-Konten ähnelt der Einstellung der Replikation, wenn beide Buckets demselben Konto gehören. Der einzige Unterschied besteht darin, dass *destination* Der Bucket-Besitzer muss das gewähren *source* Der Bucket-Besitzer hat die Erlaubnis, Objekte durch Hinzufügen einer Bucket-Richtlinie zu replizieren.

Weitere Informationen zur Konfiguration der Replikation mithilfe serverseitiger Verschlüsselung mit AWS Key Management Service in kontoübergreifenden Szenarien finden Sie unter. [Erteilen von zusätzlichen Berechtigungen für kontoübergreifende Szenarien](#)

So konfigurieren Sie die Replikation, wenn Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten

1. In diesem Beispiel erstellen Sie *source* and *destination* Eimer in zwei verschiedenen AWS-Konten. Sie müssen zwei Anmeldeinformationsprofile für die eingerichtet haben AWS CLI (In diesem Beispiel verwenden wir *acctA* und *acctB* für Profilnamen). Weitere Informationen zum Einrichten von Profilen mit Anmeldeinformationen finden Sie unter [Benannte Profile](#) im AWS Command Line Interface Benutzerleitfaden.
2. Folgen Sie den step-by-step Anweisungen unter [Konfigurieren für Buckets im selben Konto](#) mit den folgenden Änderungen:
 - Für alle AWS CLI Befehle im Zusammenhang mit *source* Bucket-Aktivitäten (zum Erstellen der *source* Bucket, Aktivierung der Versionierung und Erstellung der IAM Rolle), verwenden Sie das *acctA* Profil. Verwenden Sie das *acctB* Profil, um das zu erstellen *destination* Eimer.
 - Stellen Sie sicher, dass die Berechtigungsrichtlinie Folgendes spezifiziert *source* and *destination* Buckets, die Sie für dieses Beispiel erstellt haben.
3. Fügen Sie in der Konsole die folgende Bucket-Richtlinie auf *destination* Bucket, damit der Besitzer des *source* Bucket, um Objekte zu replizieren. Achten Sie darauf, die Richtlinie zu

bearbeiten, indem Sie Folgendes angeben AWS-Konto ID des *source* Der Besitzer des Buckets und der *destination* Name des Buckets.

Note

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen. Ersetzen *DOC-EXAMPLE-BUCKET* mit dem Namen Ihres Ziel-Buckets. Ersetzen *source-bucket-acct-ID:role/service-role/source-acct-IAM-role* mit der Rolle, die Sie für diese Replikationskonfiguration verwenden.

Wenn Sie die IAM Servicerolle manuell erstellt haben, legen Sie den Rollenpfad wie *role/service-role/* im folgenden Richtlinienbeispiel gezeigt fest. Weitere Informationen finden Sie [IAMARNs](#) im IAMBenutzerhandbuch.

```
{
  "Version":"2012-10-17",
  "Id": "",
  "Statement": [
    {
      "Sid": "Set-permissions-for-objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source-bucket-acct-ID:role/service-role/source-acct-IAM-role"
      },
      "Action": ["s3:ReplicateObject", "s3:ReplicateDelete"],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    },
    {
      "Sid": "Set permissions on bucket",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source-bucket-acct-ID:role/service-role/source-acct-IAM-role"
      },
      "Action": ["s3:GetBucketVersioning", "s3:PutBucketVersioning"],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    }
  ]
}
```

}

Wählen Sie den Bucket aus und fügen Sie die Bucket-Richtlinie hinzu. Detaillierte Anweisungen finden Sie unter [Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole](#).

Bei der Replikation besitzt der Eigentümer des Quellobjekts standardmäßig das Replikat. Wenn Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, können Sie optionale Konfigurationseinstellungen hinzufügen, um den Besitz des Replikats zu ändern AWS-Konto dem die Ziel-Buckets gehören. Dazu gehört auch die Gewährung der `ObjectOwnerOverrideToBucketOwner`-Berechtigung. Weitere Informationen finden Sie unter [Ändern des Replikat-Eigentümers](#).

Ändern des Replikat-Eigentümers

Bei der Replikation besitzt der Eigentümer des Quellobjekts standardmäßig auch das Replikat. Wenn Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten und Sie möchten den Besitz des Replikats ändern auf AWS-Konto dem die Ziel-Buckets gehören, können Sie optionale Konfigurationseinstellungen hinzufügen, um den Besitz des Replikats zu ändern AWS-Konto dem die Ziel-Buckets gehören. Sie können dies z. B. tun, um den Zugriff auf Objektreplikate einzuschränken. Dies wird auch als die Eigentümer-Überschreibungs-Option der Replikationskonfiguration bezeichnet. Weitere Informationen zur Besitzer-Überschreibungs-Option finden Sie unter [Hinzufügen der Eigentümer-Überschreibungs-Option zur Replikations-Konfiguration](#). Weitere Informationen zum Einrichten der Replikationskonfiguration finden Sie unter [Überblick über das Replizieren von Objekten](#).

Um die Eigentümer-Überschreibung zu konfigurieren, gehen Sie wie folgt vor:

- Fügen Sie die Eigentümer-Überschreibungs-Option zur Replikations-Konfiguration hinzu, um Amazon S3 anzuweisen, die Replikat-Eigentümerschaft zu ändern.
- Erteilen Sie Amazon-S3-Berechtigungen zum Ändern der Replikat-Eigentümerschaft.
- Fügen Sie in der Richtlinie für die Ziel-Buckets die Berechtigung zum Ändern der Replikat-Eigentümerschaft hinzu. So kann der Eigentümer der Ziel-Buckets die Eigentümerschaft von Objektreplikaten annehmen.

Weitere Informationen finden Sie unter [Hinzufügen der Eigentümer-Überschreibungs-Option zur Replikations-Konfiguration](#). Ein funktionierendes Beispiel mit step-by-step Anweisungen finden Sie unter [Wie ändert man den Besitzer des Replikats](#).

Vom Bucket-Eigentümer erzwungene Einstellung für Object Ownership

Wenn Sie Amazon S3 S3-Replikation verwenden und die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, kann der Bucket-Besitzer des Ziel-Buckets deaktivieren ACLs (wobei die Einstellung des Bucket-Besitzers für Objekteigentümer erzwungen wird), um den Besitz des Replikats zu ändern AWS-Konto dem der Ziel-Bucket gehört. Diese Einstellung ahmt das Verhalten der bestehenden Besitzerüberschreibung nach, ohne dass eine `s3:ObjectOwnerOverrideToBucketOwner`-Berechtigung erforderlich ist. Dies bedeutet, dass alle Objekte, die mit der erzwungenen Einstellung des Bucket-Eigentümers in den Ziel-Bucket repliziert werden, dem Eigentümer des Ziel-Buckets gehören. Informationen zu Object Ownership finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Hinzufügen der Eigentümer-Überschreibungs-Option zur Replikations-Konfiguration

Warning

Fügen Sie die Option `Owner Override` nur hinzu, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten. Amazon S3 überprüft nicht, ob die Buckets denselben oder unterschiedlichen Konten gehören. Wenn Sie den Besitzer hinzufügen, wird außer Kraft gesetzt, wenn beide Buckets demselben gehören AWS-Konto, Amazon S3 wendet die Besitzerüberschreibung an. Es gewährt dem Besitzer des Ziel-Buckets volle Rechte und repliziert keine nachfolgenden Aktualisierungen der Zugriffskontrollliste für Quellobjekte (ACL). Der Besitzer des Replikats kann die einem Replikat ACL zugewiesenen Daten mit einer `PUT ACL` Anfrage direkt ändern, jedoch nicht durch Replikation.

Um die Option zur Eigentümer-Überschreibung festzulegen, fügen Sie Folgendes zu jedem `Destination`-Element hinzu:

- Das Element `AccessControlTranslation`, das Amazon S3 anweist, die Replikateigentümerschaft zu ändern
- Das `Account` Element, das spezifiziert AWS-Konto des Besitzers des Ziel-Buckets

```
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  ...
  <Destination>
    ...
```



```

    <AccessControlTranslation>
      <Owner>Destination</Owner>
    </AccessControlTranslation>
    <Account>destination-bucket-owner-account-id</Account>
  </Destination>
</Rule>
</ReplicationConfiguration>

```

Die folgende Beispiel-Replikationskonfiguration weist Amazon S3 an, Objekte mit dem Schlüsselpräfix Tax in den Ziel-Bucket zu replizieren und die Replikateigentümerschaft zu ändern.

```

<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <ID>Rule-1</ID>
    <Priority>1</Priority>
    <Status>Enabled</Status>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <Prefix>Tax</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::destination-bucket</Bucket>
      <Account>destination-bucket-owner-account-id</Account>
      <AccessControlTranslation>
        <Owner>Destination</Owner>
      </AccessControlTranslation>
    </Destination>
  </Rule>
</ReplicationConfiguration>

```

Erteilen der Berechtigung zur Änderung der Replikat-Eigentümerschaft an Amazon S3

Gewähren Sie Amazon S3 S3-Berechtigungen zum Ändern des Replikatbesitzes, indem Sie in der mit der IAM Rolle verknüpften Berechtigungsrichtlinie die Berechtigung für die `s3:ObjectOwnerOverrideToBucketOwner` Aktion hinzufügen. Dies ist die IAM Rolle, die Sie in der Replikationskonfiguration angegeben haben und die es Amazon S3 ermöglicht, Objekte in Ihrem Namen anzunehmen und zu replizieren.

```
...
{
  "Effect":"Allow",
    "Action":[
      "s3:ObjectOwnerOverrideToBucketOwner"
    ],
  "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Hinzufügen der Berechtigung zur Ziel-Bucket-Richtlinie, um das Ändern der Replikat-Eigentümerschaft zuzulassen

Der Eigentümer des Ziel-Buckets muss dem Eigentümer des Quell-Buckets die Berechtigung zum Ändern der Replikat-Eigentümerschaft erteilen. Der Eigentümer des Ziel-Buckets erteilt dem Eigentümer des Quell-Buckets die Berechtigung für die Aktion `s3:ObjectOwnerOverrideToBucketOwner`. Dies ermöglicht dem Eigentümer des Ziel-Buckets, die Eigentümerschaft von Objektreplikaten anzunehmen. Die folgende Beispielanweisung einer Bucket-Richtlinie zeigt, wie dies funktioniert:

```
...
{
  "Sid":"1",
  "Effect":"Allow",
  "Principal":{"AWS": "source-bucket-account-id"},
  "Action":["s3:ObjectOwnerOverrideToBucketOwner"],
  "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Weitere Überlegungen

Wenn Sie die Eigentümer-Überschreibungs-Option konfigurieren, berücksichtigen Sie die folgenden Überlegungen:

- Standardmäßig besitzt der Eigentümer des Quellobjekts auch das Replikat. Amazon S3 repliziert die Objektversion und die ACL damit verbundene Version.

Wenn Sie die Besitzer-Überschreibung hinzufügen, repliziert Amazon S3 nur die Objektversion, nicht die ACL. Darüber hinaus repliziert Amazon S3 keine nachfolgenden Änderungen am

QuellobjektACL. Amazon S3 legt das ACL auf dem Replikate fest, das dem Besitzer des Ziel-Buckets die volle Kontrolle gewährt.

- Wenn Sie eine Replikations-Konfiguration ändern und die Eigentümerüberschreibung aktivieren oder deaktivieren, geschieht Folgendes:

- Wenn Sie die Eigentümerüberschreibungs-Option zur Replikations-Konfiguration hinzufügen

Wenn Amazon S3 eine Objektversion repliziert, verwirft es die Version, ACL die dem Quellobjekt zugeordnet ist. Stattdessen wird das ACL auf dem Replikate festgelegt, sodass der Besitzer des Ziel-Buckets die volle Kontrolle hat. Nachfolgende Änderungen am Quellobjekt werden nicht repliziert. Diese ACL Änderung gilt jedoch nicht für Objektversionen, die repliziert wurden, bevor Sie die Option zum Überschreiben des Besitzers festgelegt haben. ACL Aktualisierungen von Quellobjekten, die repliziert wurden, bevor die Besitzerüberschreibung festgelegt wurde, werden weiterhin repliziert (weil das Objekt und seine Replikate weiterhin denselben Besitzer haben).

- Wenn Sie die Eigentümer-Überschreibungs-Option aus der Replikations-Konfiguration entfernen

Amazon S3 repliziert neue Objekte, die im Quell-Bucket und den mit den Ziel-Buckets verknüpften ACLs Buckets erscheinen. Für Objekte, die repliziert wurden, bevor Sie die Eigentümerüberschreibung entfernt haben, repliziert Amazon S3 das nicht, ACLs da die von Amazon S3 vorgenommene Änderung des Objekteigentums weiterhin in Kraft ist. Das heißt, ACLs die Objektversion, die repliziert wurde, als die Eigentümerüberschreibung festgelegt wurde, wird weiterhin nicht repliziert.

Wie ändert man den Besitzer des Replikats

Wenn der *source* and *destination* Buckets in einer Replikationskonfiguration gehören verschiedenen AWS-Konten, können Sie Amazon S3 anweisen, den Besitz des Replikats auf das zu ändern AWS-Konto dem gehört der *destination* Eimer. Dieses Beispiel erklärt, wie Sie die Amazon S3 S3-Konsole verwenden und AWS CLI um den Besitz des Replikats zu ändern. Weitere Informationen finden Sie unter [Ändern des Replikate-Eigentümers](#).

Note

Wenn Sie die S3-Replikation verwenden und die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, kann der Bucket-Besitzer des Ziel-Buckets deaktivieren ACLs (wobei der Bucket-Besitzer die Einstellung für Objekteigentum erzwingt), um den

Besitz des Replikats zu ändern AWS-Konto dem der Ziel-Bucket gehört. Diese Einstellung ahmt das Verhalten der bestehenden Besitzerüberschreibung nach, ohne dass eine `s3:ObjectOwnerOverrideToBucketOwner`-Berechtigung erforderlich ist. Dies bedeutet, dass alle Objekte, die mit der erzwungenen Einstellung des Bucket-Eigentümers in den Ziel-Bucket repliziert werden, dem Eigentümer des Ziel-Buckets gehören. Informationen zu Object Ownership finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Weitere Informationen zur Konfiguration der Replikation mit serverseitiger Verschlüsselung finden Sie mit AWS Key Management Service in kontenübergreifenden Szenarien finden Sie unter [Erteilen von zusätzlichen Berechtigungen für kontenübergreifende Szenarien](#)

Verwenden der S3-Konsole

step-by-step Anweisungen finden Sie unter [Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos](#). Dieses Thema enthält Anweisungen zum Einrichten der Replikationskonfiguration, wenn Buckets denselben und unterschiedlichen Buckets gehören AWS-Konten.

Verwendung der AWS CLI

Um den Besitz eines Replikats zu ändern, verwenden Sie AWS CLI, Sie erstellen Buckets, aktivieren die Versionierung für die Buckets, erstellen eine IAM Rolle, die Amazon S3 die Berechtigung erteilt, Objekte zu replizieren, und fügen die Replikationskonfiguration zum Quell-Bucket hinzu. In der Replikations-Konfiguration weisen Sie Amazon S3 an, den Replikatseigentümer zu ändern. Sie testen außerdem die Einrichtung.

Um den Besitz von Replikaten zu ändern, wenn Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten (AWS CLI)

1. In diesem Beispiel erstellen Sie *source* and *destination* Eimer in zwei verschiedenen AWS-Konten Konfigurieren Sie . AWS CLI mit zwei benannten Profilen. Bei diesem Beispiel werden die Profile `acctA` bzw. `acctB` verwendet. Weitere Informationen zum Einrichten von Profilen mit Anmeldeinformationen finden Sie unter [Benannte Profile](#) in der AWS Command Line Interface Benutzerleitfaden.

⚠ Important

Die Profile, die Sie für diese Übung verwenden, müssen über die nötigen Berechtigungen verfügen. In der Replikationskonfiguration geben Sie beispielsweise die IAM Rolle an, die Amazon S3 übernehmen kann. Dies können Sie nur tun, wenn das verwendete Profil über die `iam:PassRole`-Berechtigung verfügt. Wenn Sie zur Erstellung eines benannten Profils die Benutzer-Anmeldeinformationen eines Administrators verwenden, können Sie alle Aufgaben durchführen. Weitere Informationen finden Sie unter [Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an einen AWS Service](#) im IAMBenutzerhandbuch.

Sie müssen sicherstellen, dass diese Profile über die nötigen Berechtigungen verfügen. Die Replikationskonfiguration beinhaltet beispielsweise eine IAM Rolle, die Amazon S3 übernehmen kann. Das benannte Profil, mit dem Sie eine solche Konfiguration einem Bucket anfügen, kann dies nur tun, wenn es über die `iam:PassRole`-Berechtigung verfügt. Wenn Sie zur Erstellung dieser benannten Profile die Benutzer-Anmeldeinformationen eines Administrators verwenden, haben diese alle Berechtigungen. Weitere Informationen finden Sie unter [Einem Benutzer Berechtigungen zur Übergabe einer Rolle an einen AWS Service](#) im IAMBenutzerhandbuch.

2. Erstellen Sie das *source* Bucket und aktivieren Sie die Versionierung. Dieses Beispiel erstellt die *source* Eimer in der Region USA Ost (Nord-Virginia) (us-east-1).

```
aws s3api create-bucket \  
--bucket source \  
--region us-east-1 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket source \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

3. Erstellen Sie ein *destination* Bucket und aktiviere die Versionierung. Dieses Beispiel erstellt die *destination* Eimer in der Region USA West (Oregon) (US-West-2). Verwenden Sie ein AWS-Konto Ein anderes Profil als das, das Sie für das verwendet haben *source* Eimer.

```
aws s3api create-bucket \  

```

```
--bucket destination \  
--region us-west-2 \  
--create-bucket-configuration LocationConstraint=us-west-2 \  
--profile acctB
```

```
aws s3api put-bucket-versioning \  
--bucket destination \  
--versioning-configuration Status=Enabled \  
--profile acctB
```

4. Sie müssen Ihren Berechtigungen hinzufügen *destination* Bucket-Richtlinie, die es ermöglicht, den Besitz des Replikats zu ändern.
 - a. Speichern Sie die folgende Richtlinie in *destination-bucket-policy.json*.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "destination_bucket_policy_sid",  
      "Principal": {  
        "AWS": "source-bucket-owner-account-id"  
      },  
      "Action": [  
        "s3:ReplicateObject",  
        "s3:ReplicateDelete",  
        "s3:ObjectOwnerOverrideToBucketOwner",  
        "s3:ReplicateTags",  
        "s3:GetObjectVersionTagging"  
      ],  
      "Effect": "Allow",  
      "Resource": [  
        "arn:aws:s3::destination/*"  
      ]  
    }  
  ]  
}
```

- b. Setzen Sie die obige Richtlinie auf *destination* Eimer:

```
aws s3api put-bucket-policy --region $ {destination_region} --
bucket $ {destination} --policy file://{destination_bucket_policy}.json
```

5. Erstellen Sie eine IAM Rolle. Sie geben diese Rolle in der Replikationskonfiguration an, die Sie der *source* Bucket später. Amazon S3 übernimmt diese Rolle, um Objekte in Ihrem Namen zu replizieren. Sie erstellen eine IAM Rolle in zwei Schritten:

- Erstellen Sie eine Rolle.
- Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

a. Erstellen Sie eine IAM Rolle.

- i. Kopieren Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-trust-policy.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie erteilt Amazon S3 Berechtigungen für die Übernahme der Rolle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- ii. Führen Sie Folgendes aus AWS CLI Befehl zum Erstellen einer Rolle.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json \
--profile acctA
```

b. Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

- i. Kopieren Sie die folgende Berechtigungsrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-perm-pol-changeowner.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Zugriffsrichtlinie erteilt Berechtigungen für verschiedene Amazon-S3-Bucket- und -Objektaktionen. In den folgenden Schritten erstellen Sie eine IAM Rolle und fügen diese Richtlinie der Rolle hinzu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionAcl"
      ],
      "Resource": [
        "arn:aws:s3:::source/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::source"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ObjectOwnerOverrideToBucketOwner",
        "s3:ReplicateTags",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::destination/*"
    }
  ]
}
```



```
}
```

- ii. Um eine Richtlinie zu erstellen und sie an die Rolle anzufügen, führen Sie den folgenden Befehl aus:

```
$ aws iam put-role-policy \  
--role-name replicationRole \  
--policy-document file:///s3-role-perm-pol-changeowner.json \  
--policy-name replicationRolechangeownerPolicy \  
--profile acctA
```

6. Fügen Sie Ihrem Quell-Bucket eine Replikations-Konfiguration hinzu.
 - a. Das Tool AWS CLI erfordert die Angabe der Replikationskonfiguration als JSON. Speichern Sie Folgendes JSON in einer Datei mit dem Namen `replication.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Fügen Sie in der Konfiguration `AccessControlTranslation` hinzu, um eine Änderung in der Replikateigentümerschaft anzugeben.

```
{  
  "Role": "IAM-role-ARN",  
  "Rules": [  
    {  
      "Status": "Enabled",  
      "Priority": 1,  
      "DeleteMarkerReplication": {  
        "Status": "Disabled"  
      },  
      "Filter": {  
      },  
      "Status": "Enabled",  
      "Destination": {  
        "Bucket": "arn:aws:s3:::destination",  
        "Account": "destination-bucket-owner-account-id",  
        "AccessControlTranslation": {  
          "Owner": "Destination"  
        }  
      }  
    }  
  ]  
}
```

- b. Bearbeiten Sie die, JSON indem Sie Werte für die angeben *destination* Konto-ID des Bucket-Besitzers und *IAM-role-ARN*. Speichern Sie die Änderungen.
- c. Um die Replikations-Konfiguration zum Quell-Bucket hinzuzufügen, führen Sie den folgenden Befehl aus. Geben Sie die *source* Name des Buckets.

```
$ aws s3api put-bucket-replication \  
--replication-configuration file://replication.json \  
--bucket source \  
--profile acctA
```

7. Prüfen Sie die Replikant-Eigentümerschaft in der Amazon-S3-Konsole.
 - a. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
 - b. Fügen Sie Objekte hinzu *source* Eimer. Stellen Sie sicher, dass *destination* Der Bucket enthält die Objektreplikate und der Besitzer der Replikate hat sich geändert in AWS-Konto dem gehört der *destination* Eimer.

Verwendung der AWS SDKs

Ein Code-Beispiel zum Hinzufügen einer Replikationskonfiguration finden Sie unter [Verwendung der AWS SDKs](#). Sie müssen die Replikations-Konfiguration entsprechend ändern. Weitere konzeptuelle Informationen finden Sie unter [Ändern des Replikant-Eigentümers](#).

Erfüllung der Compliance-Anforderungen mit S3 Replication Time Control (S3RTC)

S3 Replication Time Control (S3RTC) unterstützt Sie bei der Erfüllung von Compliance- oder Geschäftsanforderungen für die Datenreplikation und bietet Einblick in die Amazon S3 S3-Replikationszeiten. S3 RTC repliziert die meisten Objekte, die Sie auf Amazon S3 hochladen, innerhalb von Sekunden und 99,99 Prozent dieser Objekte innerhalb von 15 Minuten.

S3 enthält RTC standardmäßig S3-Replikationsmetriken und Amazon S3 S3-Ereignisbenachrichtigungen, mit denen Sie die Gesamtzahl der API S3-Operationen, deren Replikation aussteht, die Gesamtgröße der zu replizierenden Objekte und die maximale Replikationszeit überwachen können. Sie können Replikationsmetriken unabhängig von S3 aktivieren RTC. Weitere Informationen finden Sie unter [Überwachen des Fortschritts mit Replikationsmetriken](#). Darüber hinaus RTC bietet `OperationMissedThreshold` S3 `OperationReplicatedAfterThreshold` Ereignisse, die den Bucket-Besitzer benachrichtigen,

wenn die Objektreplikation den 15-Minuten-Schwellenwert überschreitet oder nach Ablauf dieser Frist repliziert wird.

Mit S3 können Amazon S3 S3-Ereignisse Sie in seltenen Fällen benachrichtigen RTC, wenn Objekte nicht innerhalb von 15 Minuten repliziert werden und wenn diese Objekte nach dem 15-Minuten-Schwellenwert repliziert werden. Amazon S3 S3-Ereignisse sind über Amazon SQSSNS, Amazon oder AWS Lambda. Weitere Informationen finden Sie unter [the section called “Amazon-S3-Ereignis-Benachrichtigungen”](#).

Themen

- [Steuerung der S3-Replikationszeit](#)
- [Replikationsmetriken mit S3 RTC](#)
- [Verwenden von Amazon-S3-Ereignis-Benachrichtigungen zum Nachverfolgen von Replikationsobjekten](#)
- [Bewährte Methoden und Richtlinien für S3 RTC](#)
- [Aktivieren von S3 Replication Time Control \(S3RTC\)](#)

Steuerung der S3-Replikationszeit

Sie können S3 Replication Time Control (S3RTC) mit einer neuen oder vorhandenen Replikationsregel verwenden. Sie können die Replikationsregel auf einen gesamten S3-Bucket oder auf Amazon-S3-Objekte mit einem bestimmten Präfix oder Tag anwenden. Wenn Sie S3 aktivieren RTC, werden Replikationsmetriken auch für Ihre Replikationsregel aktiviert.

Wenn Sie die aktuelle Version der Replikations-Konfiguration verwenden, d. h. das Element `Filter` in einer Replikations-Konfigurations-Regel angeben, repliziert Amazon S3 die Löschmarkierung nicht standardmäßig. Sie können den non-tag-based Regeln jedoch die Replikation von Löschmarken hinzufügen.

Note

Replikationsmetriken werden zum gleichen Tarif wie CloudWatch benutzerdefinierte Amazon-Metriken abgerechnet. Weitere Informationen finden Sie unter [CloudWatch Amazon-Preise](#).

Weitere Informationen zum Erstellen einer Regel mit S3 RTC finden Sie unter [Aktivieren von S3 Replication Time Control \(S3RTC\)](#).

Replikationsmetriken mit S3 RTC

Replikationsregeln mit aktiviertem S3 Replication Time Control (S3RTC) veröffentlichen Replikationsmetriken. Mit Replikationsmetriken können Sie die Gesamtzahl der API S3-Operationen, für die eine Replikation aussteht, die Gesamtgröße der zu replizierenden Objekte, die maximale Replizierungszeit in die Zielregion und die Gesamtzahl der Operationen, bei denen die Replikation fehlgeschlagen ist, überwachen. Anschließend können Sie jedes Dataset, das Sie separat replizieren, überwachen.

Replikationsmetriken sind innerhalb von 15 Minuten nach der Aktivierung von S3 verfügbar RTC. Replikationsmetriken sind über die [Amazon S3-Konsole](#), [Amazon S3 API](#), die AWS SDKs, der [AWS Command Line Interface \(AWS CLI\)](#) und [Amazon CloudWatch](#). Weitere Informationen finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

Weitere Informationen zum Finden von Replikationsmetriken über die Amazon-S3-Konsole finden Sie unter [Anzeigen von Replikationsmetriken](#).

Verwenden von Amazon-S3-Ereignis-Benachrichtigungen zum Nachverfolgen von Replikationsobjekten

Sie können die Replikationszeit für Objekte verfolgen, die nicht innerhalb von 15 Minuten repliziert wurden, indem Sie bestimmte Ereignisbenachrichtigungen überwachen, die S3 Replication Time Control (S3RTC) veröffentlicht. Diese Ereignisse werden veröffentlicht, wenn ein Objekt, das für die Replikation mit S3 in Frage kam, RTC nicht innerhalb von 15 Minuten repliziert wurde, und wenn dieses Objekt nach dem Schwellenwert von 15 Minuten repliziert wird.

Replikationsereignisse sind innerhalb von 15 Minuten nach Aktivierung von S3 verfügbar. RTC Amazon S3 S3-Ereignisse sind über Amazon SQSSNS, Amazon oder AWS Lambda. Weitere Informationen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Bewährte Methoden und Richtlinien für S3 RTC

Wenn Sie Daten in Amazon S3 mit S3 Replication Time Control (S3RTC) replizieren, befolgen Sie diese Best-Practice-Richtlinien, um die Replikationsleistung für Ihre Workloads zu optimieren.

Themen

- [Leistungsrichtlinien für Amazon-S3-Replikation and -Anforderungsraten](#)
- [Schätzen der Replikationsanforderungsraten](#)
- [Überschreitung der RTC S3-Datenübertragungsratenlimits](#)

- [AWS KMS Anforderungsraten für die Replikation verschlüsselter Objekte](#)

Leistungsrichtlinien für Amazon-S3-Replikation and -Anforderungsraten

Wenn Speicherinhalte zu Amazon S3 hochgeladen oder von dort abgerufen werden, können Ihre Anwendungen Tausende von Transaktionen pro Sekunde bei der Anforderungsleistung erhalten. Beispielsweise kann eine Anwendung mindestens 3.500 PUT//DELETE oder 5.500 COPY POSTGET//HEAD Anfragen pro Sekunde pro Präfix in einem S3-Bucket erreichen, einschließlich der Anfragen, die die S3-Replikation in Ihrem Namen stellt. Es gibt keine Einschränkungen für die Anzahl der Präfixe in einem Bucket. Sie können Ihre Lese- und Schreibleistung steigern, indem Sie Lesevorgänge parallelisieren. Wenn Sie beispielsweise 10 Präfixe in einem S3-Bucket für parallele Lesevorgänge einrichten, können Sie damit die Leseleistung auf 55 000 Leseanfragen pro Sekunde skalieren.

Amazon S3 skaliert automatisch als Reaktion auf anhaltende Anforderungsraten, die über diesen Richtlinien liegen, oder auf anhaltende Anforderungsraten, die gleichzeitig mit LIST Anfragen einhergehen. Während Amazon S3 intern für die neue Anforderungsrate optimiert, erhalten Sie möglicherweise vorübergehend HTTP 503 Antworten auf Anfragen, bis die Optimierung abgeschlossen ist. Dies kann bei steigenden Anforderungsraten pro Sekunde oder bei der ersten Aktivierung von S3 der Fall sein. Während dieser Zeiträume kann sich die Replikationslatenz erhöhen. Das RTC S3-Service Level Agreement (SLA) gilt nicht für Zeiträume, in denen die Amazon S3 S3-Leistungsrichtlinien für Anfragen pro Sekunde überschritten werden.

S3 gilt RTC SLA auch nicht in Zeiträumen, in denen Ihre Replikationsdatenübertragungsrate das Standardlimit von 1 Gbit/s überschreitet. Wenn Sie davon ausgehen, dass Ihre Replikationsübertragungsrate 1 Gbit/s überschreiten wird, wenden Sie sich an [AWS Support Zentrieren Sie](#) oder verwenden Sie [Service Quotas](#), um eine Erhöhung Ihres Limits zu beantragen.

Schätzen der Replikationsanforderungsraten

Ihre Gesamtanforderungsrate einschließlich der Anforderungen, die die Amazon-S3-Replikation in Ihrem Namen vornimmt, sollte den Richtlinien für die Amazon-S3-Anforderungsrate sowohl für die Replikationsquelle als auch für die Ziel-Buckets entsprechen. Für jedes replizierte Objekt stellt die Amazon S3 S3-Replikation bis zu fünf GET HEAD /-Anfragen und eine PUT Anfrage an den Quell-Bucket und eine PUT Anfrage an jeden Ziel-Bucket.

Wenn Sie beispielsweise erwarten, 100 Objekte pro Sekunde zu replizieren, kann die Amazon S3 S3-Replikation in Ihrem Namen weitere 100 PUT Anfragen, also insgesamt 200 PUTs pro Sekunde,

an den S3-Quell-Bucket ausführen. Die Amazon S3 S3-Replikation kann auch bis zu 500GET/HEAD(5GET//HEADAnfragen für jedes replizierte Objekt) ausführen.

Note

Es fallen Kosten für nur eine PUT Anfrage pro repliziertem Objekt an. Weitere Informationen finden Sie in den Preisinformationen im [Amazon S3 FAQ zur Replikation](#).

Überschreitung der RTC S3-Datenübertragungsratenlimits

Wenn Sie davon ausgehen, dass Ihre S3 Replication Time Control-Datenübertragungsrate das Standardlimit von 1 Gbit/s überschreitet, wenden Sie sich an [AWS Support Zentrieren Sie](#) oder verwenden Sie [Service Quotas](#), um eine Erhöhung Ihres Limits zu beantragen.

AWS KMS Anforderungsraten für die Replikation verschlüsselter Objekte

Wenn Sie mit serverseitiger Verschlüsselung (SSE-KMS) verschlüsselte Objekte mithilfe der Amazon S3 S3-Replikation replizieren, AWS Key Management Service (AWS KMS) Es gelten Grenzwerte für Anfragen pro Sekunde. AWS KMS könnte eine ansonsten gültige Anfrage ablehnen, weil Ihre Anforderungsrate das Limit für die Anzahl der Anfragen pro Sekunde überschreitet. Wenn eine Anfrage gedrosselt wird, AWS KMS gibt einen `ThrottlingException` Fehler zurück. Das Tool AWS KMS Das Anforderungsratenlimit gilt für Anfragen, die Sie direkt stellen, und für Anfragen, die von der Amazon S3 S3-Replikation in Ihrem Namen gestellt werden.

Wenn Sie beispielsweise damit rechnen, 1.000 Objekte pro Sekunde zu replizieren, können Sie 2.000 Anfragen von Ihrem AWS KMS Begrenzung der Anforderungsrate. Die daraus resultierende Anforderungsrate pro Sekunde ist für Ihre verfügbar AWS KMS Workloads ohne Replikation. Sie können verwenden [AWS KMS Metriken in Amazon anfordern CloudWatch](#), um die Gesamtsumme zu überwachen AWS KMS fordern Sie den Tarif auf Ihrem an AWS-Konto.

Aktivieren von S3 Replication Time Control (S3RTC)

S3 Replication Time Control (S3RTC) unterstützt Sie bei der Erfüllung von Compliance- oder Geschäftsanforderungen für die Datenreplikation und bietet Einblick in die Amazon S3 S3-Replikationszeiten. S3 RTC repliziert die meisten Objekte, die Sie auf Amazon S3 hochladen, innerhalb von Sekunden und 99,99 Prozent dieser Objekte innerhalb von 15 Minuten.

Mit S3 können Sie die Gesamtzahl und Größe der ObjekteRTC, deren Replikation aussteht, sowie die maximale Replikationszeit in der Zielregion überwachen. Replikationsmetriken sind verfügbar über

[AWS Management Console](#) und [CloudWatch Amazon-Benutzerhandbuch](#). Weitere Informationen finden Sie unter [the section called "Metriken zur S3-Replikation in CloudWatch"](#).

Verwenden der S3-Konsole

step-by-step Anweisungen finden Sie unter [Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos](#). Dieses Thema enthält Anweisungen zur Aktivierung von S3 RTC in Ihrer Replikationskonfiguration, wenn Buckets denselben und unterschiedlichen Buckets gehören AWS-Konten.

Verwendung der AWS CLI

Um das AWS CLI Um Objekte mit RTC aktiviertem S3 zu replizieren, erstellen Sie Buckets, aktivieren die Versionierung für die Buckets, erstellen eine IAM Rolle, die Amazon S3 die Berechtigung erteilt, Objekte zu replizieren, und fügen die Replikationskonfiguration zum Quell-Bucket hinzu. Für die Replikationskonfiguration muss S3 Replication Time Control (S3) aktiviert sein. RTC

Um bei RTC aktiviertem S3 zu replizieren (AWS CLI)

- Im folgenden Beispiel werden `ReplicationTime` und `Metric` festgelegt und die Replikationskonfiguration zum Quell-Bucket hinzugefügt.

```
{
  "Rules": [
    {
      "Status": "Enabled",
      "Filter": {
        "Prefix": "Tax"
      },
      "DeleteMarkerReplication": {
        "Status": "Disabled"
      },
      "Destination": {
        "Bucket": "arn:aws:s3:::destination",
        "Metrics": {
          "Status": "Enabled",
          "EventThreshold": {
            "Minutes": 15
          }
        },
        "ReplicationTime": {
          "Status": "Enabled",
```

```
        "Time": {
            "Minutes": 15
        }
    },
    "Priority": 1
}
],
"Role": "IAM-Role-ARN"
}
```

Important

Für `Metrics:EventThreshold:Minutes` und `ReplicationTime:Time:Minutes` ist als gültiger Wert nur 15 zulässig.

Verwendung der AWS SDK für Java

Das folgende Java-Beispiel fügt die Replikationskonfiguration mit S3 Replication Time Control (S3RTC) hinzu.

```
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.DeleteMarkerReplication;
import software.amazon.awssdk.services.s3.model.Destination;
import software.amazon.awssdk.services.s3.model.Metrics;
import software.amazon.awssdk.services.s3.model.MetricsStatus;
import software.amazon.awssdk.services.s3.model.PutBucketReplicationRequest;
import software.amazon.awssdk.services.s3.model.ReplicationConfiguration;
import software.amazon.awssdk.services.s3.model.ReplicationRule;
import software.amazon.awssdk.services.s3.model.ReplicationRuleFilter;
import software.amazon.awssdk.services.s3.model.ReplicationTime;
import software.amazon.awssdk.services.s3.model.ReplicationTimeStatus;
import software.amazon.awssdk.services.s3.model.ReplicationTimeValue;

public class Main {

    public static void main(String[] args) {
        S3Client s3 = S3Client.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(() -> AwsBasicCredentials.create(
```



```
        "AWS_ACCESS_KEY_ID",
        "AWS_SECRET_ACCESS_KEY")
    )
    .build();

ReplicationConfiguration replicationConfig = ReplicationConfiguration
    .builder()
    .rules(
        ReplicationRule
            .builder()
            .status("Enabled")
            .priority(1)
            .deleteMarkerReplication(
                DeleteMarkerReplication
                    .builder()
                    .status("Disabled")
                    .build()
            )
            .destination(
                Destination
                    .builder()
                    .bucket("destination_bucket_arn")
                    .replicationTime(
                        ReplicationTime.builder().time(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
                            ReplicationTimeStatus.ENABLED
                        ).build()
                    )
                    .metrics(
                        Metrics.builder().eventThreshold(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
                            MetricsStatus.ENABLED
                        ).build()
                    )
                    .build()
            )
            .filter(
                ReplicationRuleFilter
                    .builder()
                    .prefix("testtest")
                    .build()
            )
    )
    .build();
```

```
        .build())
        .role("role_arn")
        .build();

// Put replication configuration
PutBucketReplicationRequest putBucketReplicationRequest =
PutBucketReplicationRequest
    .builder()
    .bucket("source_bucket")
    .replicationConfiguration(replicationConfig)
    .build();

s3.putBucketReplication(putBucketReplicationRequest);
}
}
```

Weitere Informationen finden Sie unter [Erfüllung der Compliance-Anforderungen mit S3 Replication Time Control \(S3RTC\)](#).

Replikation verschlüsselter Objekte (SSE-C, SSE -S3, -, SSE -KMS) DSSE KMS

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Bei der Replikation von Objekten, die mit der serverseitigen Verschlüsselung verschlüsselt wurden, sind einige besondere Punkte zu beachten. Amazon S3 unterstützt die folgenden Arten von serverseitiger Verschlüsselung:

- Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)

- Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS)
- Zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (-) DSSE KMS
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Weitere Informationen zur serverseitigen Verschlüsselung finden Sie unter [the section called “Server-side encryption”](#).

In diesem Thema werden die Berechtigungen erläutert, die Sie benötigen, um Amazon S3 anzuweisen, Objekte zu replizieren, die mithilfe serverseitiger Verschlüsselung verschlüsselt wurden. Dieses Thema enthält auch zusätzliche Konfigurationselemente, die Sie hinzufügen können, sowie Beispiele AWS Identity and Access Management (IAM) Richtlinien, die die erforderlichen Berechtigungen für die Replikation verschlüsselter Objekte gewähren.

Ein Beispiel mit step-by-step Anweisungen finden Sie unter [Aktivierung der Replikation für verschlüsselte Objekte](#). Weitere Informationen zum Erstellen einer Replikationskonfiguration finden Sie unter [Überblick über das Replizieren von Objekten](#).

Note

Sie können Multiregion verwenden AWS KMS keys auf Amazon S3. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter [Verwenden von Schlüsseln für mehrere Regionen in AWS Key Management Service Entwicklerhandbuch](#).

Themen

- [So wirkt sich die Standard-Bucket-Verschlüsselung auf die Replikation aus](#)
- [Replizieren von Objekten, die mit -C verschlüsselt wurden SSE](#)
- [Replizieren von Objekten, die mit SSE -S3, SSE - oder - verschlüsselt wurden KMS DSSE KMS](#)
- [Aktivierung der Replikation für verschlüsselte Objekte](#)

So wirkt sich die Standard-Bucket-Verschlüsselung auf die Replikation aus

Wenn Sie die Standard-Verschlüsselung für einen Replikations-Ziel-Bucket aktivieren, gilt das folgende Verschlüsselungsverhalten:

- Wenn Objekte im Quell-Bucket nicht verschlüsselt sind, werden die Replikatobjekte im Ziel-Bucket mithilfe der Einstellungen der Standard-Verschlüsselung des Ziel-Buckets verschlüsselt. Infolgedessen unterscheiden sich die Entity-Tags (ETags) der Quellobjekte ETags von denen der Replikatobjekte. Wenn Sie Anwendungen haben, die dies verwenden ETags, müssen Sie diese Anwendungen aktualisieren, um diesem Unterschied Rechnung zu tragen.
- Wenn Objekte im Quell-Bucket mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt werden, serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS), die Replikatobjekte im Ziel-Bucket verwenden denselben Verschlüsselungstyp wie die Quellobjekte. Die Einstellungen der Standard-Verschlüsselung des Ziel-Buckets werden nicht verwendet.

Replizieren von Objekten, die mit -C verschlüsselt wurden SSE

Durch die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) können Sie Ihre eigenen proprietären Verschlüsselungsschlüssel verwalten. Mit SSE -C verwalten Sie die Schlüssel, während Amazon S3 den Verschlüsselungs- und Entschlüsselungsprozess verwaltet. Sie müssen einen Verschlüsselungsschlüssel als Teil Ihrer Anfrage angeben, brauchen aber keinen Code zu schreiben, um die Objektverschlüsselung oder -entschlüsselung durchzuführen. Wenn Sie ein Objekt hochladen, verschlüsselt Amazon S3 das Objekt mithilfe des von Ihnen angegebenen Schlüssels. Amazon S3 löscht diesen Schlüssel dann aus dem Speicher. Wenn Sie ein Objekt abrufen, müssen Sie denselben Verschlüsselungsschlüssel als Teil Ihrer Anfrage angeben. Weitere Informationen finden Sie unter [the section called “Vom Kunden bereitgestellte Verschlüsselungsschlüssel \(-C\) SSE”](#).

Die S3-Replikation unterstützt Objekte, die mit SSE -C verschlüsselt sind. Sie können die SSE -C-Objektreplikation in der Amazon S3 S3-Konsole oder mit dem AWS SDKs, genauso wie Sie die Replikation für unverschlüsselte Objekte konfigurieren. Es gibt keine zusätzlichen SSE -C-Berechtigungen, die über das hinausgehen, was derzeit für die Replikation erforderlich ist.

S3 Replication repliziert automatisch neu hochgeladene SSE -C-verschlüsselte Objekte, sofern sie geeignet sind, wie in Ihrer S3-Replikationskonfiguration angegeben. Verwenden Sie die S3-Batchreplikation, um vorhandene Objekte in Ihren Buckets zu replizieren. Weitere Informationen zum Replizieren von Objekten finden Sie unter [the section called “Live-Replikation einrichten”](#) und [the section called “Replizieren vorhandener Objekte”](#).

Für die Replikation SSE von -C-Objekten fallen keine zusätzlichen Gebühren an. Einzelheiten zu den Replikationspreisen finden Sie auf der Seite [Amazon S3 – Preise](#).

Replizieren von Objekten, die mit SSE -S3, SSE - oder - verschlüsselt wurden KMS DSSE KMS

Standardmäßig repliziert Amazon S3 keine Objekte, die mit SSE - KMS oder DSSE - KMS verschlüsselt sind. In diesem Abschnitt werden die zusätzlichen Konfigurationselemente erörtert, die Sie hinzufügen können, um Amazon S3 anzuweisen, diese Objekte zu replizieren.

Ein Beispiel mit step-by-step Anweisungen finden Sie unter [Aktivierung der Replikation für verschlüsselte Objekte](#). Weitere Informationen zum Erstellen einer Replikationskonfiguration finden Sie unter [Überblick über das Replizieren von Objekten](#).

Angeben zusätzlicher Informationen in der Replikations-Konfiguration

In der Replikations-Konfiguration machen Sie Folgendes:

- Fügen Sie dem `Destination` Element in Ihrer Replikationskonfiguration die ID des symmetrischen Elements hinzu AWS KMS vom Kunden verwalteter Schlüssel, den Amazon S3 zur Verschlüsselung von Objektreplikaten verwenden soll, wie in der folgenden Beispiel-Replikationskonfiguration dargestellt.
- Entscheiden Sie sich ausdrücklich dafür, indem Sie die Replikation von Objekten aktivieren, die mithilfe von KMS Schlüsseln (SSE- KMS oder DSSE -KMS) verschlüsselt wurden. Fügen Sie dazu das `SourceSelectionCriteria`-Element hinzu, wie im folgenden Beispiel einer Replikationskonfiguration dargestellt.

```
<ReplicationConfiguration>
  <Rule>
    ...
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>

    <Destination>
      ...
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same AWS-
        Region as the destination bucket.</ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

```

    ...
  </Rule>
</ReplicationConfiguration>

```

Important

Der KMS Schlüssel muss in derselben Datei erstellt worden sein AWS-Region wie die Ziel-Buckets.

Der KMS Schlüssel muss gültig sein. Die PutBucketReplication API Operation überprüft nicht die Gültigkeit von KMS Schlüsseln. Wenn Sie einen ungültigen KMS Schlüssel verwenden, erhalten Sie als Antwort den HTTP 200 OK Statuscode, aber die Replikation schlägt fehl.

Das folgende Beispiel zeigt eine Replikationskonfiguration, die optionale Konfigurationselemente enthält. Diese Replikations-Konfiguration weist eine Regel auf. Die Regel wird auf alle Objekte mit dem Schlüsselpräfix Tax angewendet. Amazon S3 verwendet die angegebenen AWS KMS key ID zum Verschlüsseln dieser Objektreplicate.

```

<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <ID>Rule-1</ID>
    <Priority>1</Priority>
    <Status>Enabled</Status>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <Prefix>Tax</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3::amzn-s3-demo-destination-bucket</Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same AWS-Region as the destination bucket. (S3 uses this key to encrypt object replicas.)</ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
  </Rule>
</ReplicationConfiguration>

```

```
<SseKmsEncryptedObjects>
  <Status>Enabled</Status>
</SseKmsEncryptedObjects>
</SourceSelectionCriteria>
</Rule>
</ReplicationConfiguration>
```

Erteilen zusätzlicher Berechtigungen für die Rolle IAM

Um Objekte zu replizieren, die im Ruhezustand mit SSE -S3KMS, SSE - oder DSSE - verschlüsselt wurden, gewähren Sie dem AWS Identity and Access Management (IAM) Rolle, die Sie in der Replikationskonfiguration angeben. Sie gewähren diese Berechtigungen, indem Sie die der IAM Rolle zugeordnete Berechtigungsrichtlinie aktualisieren.

- **s3:GetObjectVersionForReplication** Aktion für Quellobjekte — Diese Aktion ermöglicht Amazon S3, sowohl unverschlüsselte Objekte als auch Objekte zu replizieren, die mit serverseitiger Verschlüsselung mithilfe von SSE -S3, SSE - oder - KMS erstellt wurden. DSSE KMS

Note

Wir empfehlen, dass Sie die Aktion `s3:GetObjectVersionForReplication` statt der Aktion `s3:GetObjectVersion` verwenden, da `s3:GetObjectVersionForReplication` Amazon S3 nur die minimalen Berechtigungen bereitstellt, die für eine Replikation nötig sind. Darüber hinaus ermöglicht die `s3:GetObjectVersion` Aktion die Replikation von unverschlüsselten und SSE -S3-verschlüsselten Objekten, jedoch nicht von Objekten, die mithilfe von Schlüsseln (- oder -) verschlüsselt wurden. KMS SSE KMS DSSE KMS

- **kms:Decrypt** und **kms:Encrypt** AWS KMS Aktionen für die Schlüssel KMS
 - Sie müssen `kms:Decrypt` Berechtigungen erteilen für AWS KMS key das wird verwendet, um das Quellobjekt zu entschlüsseln.
 - Sie müssen `kms:Encrypt` Berechtigungen erteilen für AWS KMS key das wird verwendet, um das Objektrepikat zu verschlüsseln.
- **kms:GenerateDataKey** Aktion zum Replizieren von Klartextobjekten — Wenn Sie Klartextobjekte in einen Bucket replizieren, bei dem SSE die KMS Verschlüsselung standardmäßig aktiviert KMS oder DSSE aktiviert ist, müssen Sie die `kms:GenerateDataKey` Berechtigung für den Zielverschlüsselungskontext und den Schlüssel in die Richtlinie aufnehmen. KMS IAM

Wir empfehlen, diese Berechtigungen nur auf die Ziel-Buckets und -Objekte zu beschränken, indem Sie AWS KMS Bedingungsschlüssel. Das Tool AWS-Konto dem die IAM Rolle gehört, muss über Berechtigungen für die Schlüssel `kms:Encrypt` und `kms:Decrypt` Aktionen für die KMS Schlüssel verfügen, die in der Richtlinie aufgeführt sind. Wenn die KMS Schlüssel einem anderen gehören AWS-Konto, muss der Besitzer der KMS Schlüssel diese Rechte dem erteilen AWS-Konto dem die IAM Rolle gehört. Weitere Informationen zur Verwaltung des Zugriffs auf diese KMS Schlüssel finden Sie unter IAM Richtlinien [verwenden mit AWS KMS](#) in der AWS Key Management Service Leitfaden für Entwickler.

S3 Bucket-Schlüssel und Replikation

Um die Replikation mit einem S3-Bucket Key zu verwenden, AWS KMS key Die Richtlinie für den KMS Schlüssel, der zur Verschlüsselung des Objektreplikats verwendet wird, muss die `kms:Decrypt` Berechtigung für den aufrufenden Prinzipal enthalten. Der Aufruf zur `kms:Decrypt` Überprüfung der Integrität des S3-Bucket-Schlüssels, bevor er verwendet wird. Weitere Informationen finden Sie unter [Verwenden eines S3-Bucket-Schlüssels mit Replikation](#).

Wenn ein S3-Bucket Key für den Quell- oder Ziel-Bucket aktiviert ist, ist der Verschlüsselungskontext der Amazon-Ressourcenname (ARN) des Buckets, nicht der des Objekts ARN (zum Beispielaarn:aws:s3:::*bucket_ARN*). Sie müssen Ihre IAM Richtlinien aktualisieren, um den Bucket ARN für den Verschlüsselungskontext verwenden zu können:

```
"kms:EncryptionContext:aws:s3:arn": [  
  "arn:aws:s3:::bucket_ARN"  
]
```

Weitere Informationen finden Sie unter [Verschlüsselungskontext \(x-amz-server-side-encryption-context\)](#) (im Abschnitt „Verwenden von RESTAPI“) und [Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten](#).

Beispielrichtlinien — Verwendung von SSE -S3 und SSE — KMS mit Replikation

Die folgenden IAM Beispielrichtlinien zeigen Anweisungen für die Verwendung von SSE -S3 und SSE - KMS bei Replikation.

Example — Verwenden von SSE - KMS mit separaten Ziel-Buckets

Die folgende Beispielrichtlinie zeigt Anweisungen zur Verwendung von SSE - KMS mit separaten Ziel-Buckets.

```
{
```



```

"Version": "2012-10-17",
"Statement": [
  {
    "Action": ["kms:Decrypt"],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
        ]
      }
    },
    "Resource": [
      "List of AWS KMS key ARNs that are used to encrypt source objects."
    ]
  },
  {
    "Action": ["kms:Encrypt"],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.destination-bucket-1-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3:::amzn-s3-demo-destination-bucket1/key-prefix1*"
        ]
      }
    },
    "Resource": [
      "AWS KMS key ARNs (in the same AWS-Region as destination bucket 1). Used to encrypt object replicas created in destination bucket 1."
    ]
  },
  {
    "Action": ["kms:Encrypt"],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.destination-bucket-2-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3:::amzn-s3-demo-destination-bucket2/key-prefix1*"
        ]
      }
    }
  }
],

```

```

    "Resource": [
      "AWS KMS key ARNs (in the same AWS-Region as destination bucket 2). Used to encrypt object replicas created in destination bucket 2."
    ]
  }
]
}

```

Example — Replizieren von Objekten, die mit SSE -S3 und - erstellt wurden SSE KMS

Im Folgenden finden Sie eine vollständige IAM Richtlinie, die die erforderlichen Berechtigungen zum Replizieren von unverschlüsselten Objekten, Objekten, die mit SSE -S3 erstellt wurden, und Objekten, die mit - erstellt wurden, gewährt. SSE KMS

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:GetReplicationConfiguration",
        "s3:ListBucket"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-source-bucket"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionAcl"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:ReplicateObject",
        "s3:ReplicateDelete"
      ],
    }
  ]
}

```

```

    "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/key-prefix1*",
  },
  {
    "Action": [
      "kms:Decrypt"
    ],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
        ]
      }
    },
    "Resource": [
      "List of the AWS KMS key ARNs that are used to encrypt source objects."
    ]
  },
  {
    "Action": [
      "kms:Encrypt"
    ],
    "Effect": "Allow",
    "Condition": {
      "StringLike": {
        "kms:ViaService": "s3.destination-bucket-region.amazonaws.com",
        "kms:EncryptionContext:aws:s3:arn": [
          "arn:aws:s3:::amzn-s3-demo-destination-bucket/prefix1*"
        ]
      }
    },
    "Resource": [
      "AWS KMS key ARNs (in the same AWS-Region as the destination bucket) to use for encrypting object replicas"
    ]
  }
]
}

```

Example – Replizieren von Objekten mit S3-Bucket-Schlüsseln

Im Folgenden finden Sie eine vollständige IAM Richtlinie, die die erforderlichen Berechtigungen zum Replizieren von Objekten mit S3-Bucket-Keys gewährt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetReplicationConfiguration",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-source-bucket"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/key-prefix1*"
    },
    {
      "Action": [
        "kms:Decrypt"
      ],
      "Effect": "Allow",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "s3.source-bucket-region.amazonaws.com",

```

```

        "kms:EncryptionContext:aws:s3:arn": [
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
        ]
    },
    "Resource": [
        "List of the AWS KMS key ARNs that are used to encrypt source objects."
    ],
    {
        "Action": [
            "kms:Encrypt"
        ],
        "Effect": "Allow",
        "Condition": {
            "StringLike": {
                "kms:ViaService": "s3.destination-bucket-region.amazonaws.com",
                "kms:EncryptionContext:aws:s3:arn": [
                    "arn:aws:s3:::amzn-s3-demo-destination-bucket"
                ]
            }
        },
        "Resource": [
            "AWS KMS key ARNs (in the same AWS-Region as the destination bucket) to use
            for encrypting object replicas"
        ]
    }
]
}

```

Erteilen von zusätzlichen Berechtigungen für kontenübergreifende Szenarien

In einem kontoübergreifenden Szenario, in dem die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, können Sie einen KMS Schlüssel verwenden, um Objektreplikate zu verschlüsseln. Der KMS Schlüsselbesitzer muss dem Besitzer des Quell-Buckets jedoch die Erlaubnis zur Verwendung des KMS Schlüssels erteilen.

Note

Wenn Sie KMS Daten kontoübergreifend SSE replizieren müssen, muss Ihre Replikationsregel einen vom [Kunden verwalteten](#) Schlüssel von angeben AWS KMS für das

Zielkonto. [Von AWS verwaltete Schlüsselerlauben](#) keine kontoübergreifende Verwendung und können daher nicht für die kontenübergreifende Replikation verwendet werden.

Um dem Besitzer des Quell-Buckets die Erlaubnis zur Verwendung des KMS Schlüssels zu erteilen (AWS KMS Konsole)

1. Loggen Sie sich ein in AWS Management Console und öffne das AWS KMS Konsole bei <https://console.aws.amazon.com/kms>.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus.
4. Wählen Sie den KMS-Schlüssel aus.
5. Wählen Sie unter Allgemeine Konfiguration den Tab Schlüsselrichtlinie aus.
6. Scrollen Sie nach unten zu „Andere“ AWS-Konten.
7. Wählen Sie Andere hinzufügen AWS-Konten.

Der Andere AWS-Konten Das Dialogfeld wird angezeigt.

8. Wählen Sie im Dialogfeld die Option Weitere hinzufügen AWS-Konto. Für arn:aws:iam::In geben Sie die Konto-ID des Quell-Buckets ein.
9. Wählen Sie Änderungen speichern.

Um dem Besitzer des Quell-Buckets die Erlaubnis zur Verwendung des KMS Schlüssels zu erteilen (AWS CLI)

- Für Informationen über `put-key-policy` AWS Command Line Interface (AWS CLI) -Befehl finden Sie unter [put-key-policy](#) in der AWS CLI Befehlsreferenz. Hinweise zur zugrunde liegenden PutKeyPolicy API Operation finden Sie unter [PutKeyPolicy](#) in der [AWS Key Management Service APIReferenz](#).

AWS KMS Überlegungen zu Transaktionsquoten

Wenn Sie viele neue Objekte hinzufügen mit AWS KMS Verschlüsselung nach der Aktivierung der regionsübergreifenden Replikation (CRR) kann es zu Drosselungen (HTTP503 Service

UnavailableFehlern) kommen. Eine Drosselung tritt auf, wenn die Anzahl der AWS KMS Transaktionen pro Sekunde überschreiten das aktuelle Kontingent. Weitere Informationen finden Sie unter [Kontingente](#) im AWS Key Management Service Leitfaden für Entwickler.

Zum Anfordern einer Erhöhung für ein Kontingent verwenden Sie Service-Quotas verwenden. Weitere Informationen finden Sie unter [Anfordern einer Kontingenterhöhung](#). Wenn Service Quotas in Ihrer Region nicht unterstützt werden, [öffnen Sie eine AWS Support Fall](#).

Aktivierung der Replikation für verschlüsselte Objekte

Standardmäßig repliziert Amazon S3 keine Objekte, die mit serverseitiger Verschlüsselung verschlüsselt wurden mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüssel (DSSE-KMS). Um Objekte zu replizieren, die mit SSE - KMS oder DSS - verschlüsselt wurdenKMS, müssen Sie die Bucket-Replikationskonfiguration so ändern, dass Amazon S3 diese Objekte replizieren soll. Dieses Beispiel erklärt, wie Sie die Amazon S3 S3-Konsole verwenden und AWS Command Line Interface (AWS CLI), um die Konfiguration der Bucket-Replikation so zu ändern, dass verschlüsselte Objekte repliziert werden können.

Weitere Informationen finden Sie unter [Replikation verschlüsselter Objekte \(SSE-C, SSE -S3, -, SSE -KMS\) DSSE KMS](#).

Note

Wenn ein S3-Bucket Key für den Quell- oder Ziel-Bucket aktiviert ist, ist der Verschlüsselungskontext der Amazon-Ressourcename (ARN) des Buckets, nicht der des ObjektsARN. Sie müssen Ihre IAM Richtlinien aktualisieren, um den Bucket ARN für den Verschlüsselungskontext verwenden zu können. Weitere Informationen finden Sie unter [S3 Bucket-Schlüssel und Replikation](#).

Note

Sie können Multiregion verwenden AWS KMS keys auf Amazon S3. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter [Verwenden von Schlüsseln für mehrere Regionen in](#) AWS Key Management Service Entwicklerhandbuch.

Verwenden der S3-Konsole

step-by-step Anweisungen finden Sie unter [Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos](#). Dieses Thema enthält Anweisungen zum Einrichten einer Replikationskonfiguration, wenn die Buckets denselben und unterschiedlichen Buckets gehören AWS-Konten.

Verwendung der AWS CLI

Um verschlüsselte Objekte zu replizieren mit AWS CLI, gehen Sie wie folgt vor:

- Erstellen Sie Quell- und Ziel-Buckets und aktivieren Sie die Versionsverwaltung für diese Buckets.
- Erstellen Sie ein AWS Identity and Access Management (IAM) Servicerolle, die Amazon S3 die Erlaubnis erteilt, Objekte zu replizieren. Die Berechtigungen der IAM Rolle beinhalten die erforderlichen Berechtigungen, um die verschlüsselten Objekte zu replizieren.
- Fügen Sie eine Replikationskonfiguration zum Quell-Bucket hinzu. Die Replikationskonfiguration enthält Informationen zur Replikation von Objekten, die mithilfe KMS von Schlüsseln verschlüsselt wurden.
- Fügen Sie verschlüsselte Objekte zum Quell-Bucket hinzu.
- Testen Sie die Einrichtung, um sicherzustellen, dass Ihre verschlüsselten Objekte in den Ziel-Bucket repliziert werden.

Die folgenden Verfahren führen Sie durch diesen Prozess.

Um serverseitig verschlüsselte Objekte zu replizieren (AWS CLI)

1. In diesem Beispiel erstellen Sie sowohl *amzn-s3-demo-source-bucket* and *amzn-s3-demo-destination-bucket* Eimer im selben AWS-Konto Sie richten auch das Anmeldeinformationsprofil für die ein. AWS CLI. In diesem Beispiel wird der Profilname verwendet *acctA*.

Weitere Informationen zum Einrichten von Profilen mit Anmeldeinformationen finden Sie unter [Benannte Profile](#) in der AWS Command Line Interface Benutzerleitfaden. Die Befehle in diesem Beispiel können Sie verwenden, indem Sie die *user input placeholders* durch Ihre eigenen Informationen ersetzen.

2. Verwenden Sie die folgenden Befehle, um das zu erstellen *DOC-EXAMPLE-SOURCE-BUCKET* Bucket und aktivieren Sie die Versionierung darauf. Die folgenden Beispielbefehle erstellen *DOC-EXAMPLE-SOURCE-BUCKET* Bucket in der Region USA Ost (Nord-Virginia - east-1) ().


```
aws s3api create-bucket \  
--bucket DOC-EXAMPLE-SOURCE-BUCKET \  
--region us-east-1 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket DOC-EXAMPLE-SOURCE-BUCKET \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

3. Verwenden Sie die folgenden Befehle, um das zu erstellen *DOC-EXAMPLE-DESTINATION-BUCKET* Bucket und aktivieren Sie die Versionierung darauf. Die folgenden Beispielbefehle erstellen *DOC-EXAMPLE-DESTINATION-BUCKET* Bucket in der Region USA West (Oregonus-west-2) ().

Note

Um eine Replikationskonfiguration einzurichten, wenn beide *DOC-EXAMPLE-SOURCE-BUCKET* und *DOC-EXAMPLE-DESTINATION-BUCKET* Buckets befinden sich im selben AWS-Konto, du verwendest dasselbe Profil. In diesem Beispiel verwenden wir *acctA*. Um die Replikation zu konfigurieren, wenn die Buckets unterschiedlichen Besitzern gehören AWS-Konten, geben Sie für jedes Profil unterschiedliche Profile an.

```
aws s3api create-bucket \  
--bucket DOC-EXAMPLE-DESTINATION-BUCKET \  
--region us-west-2 \  
--create-bucket-configuration LocationConstraint=us-west-2 \  
--profile acctA
```

```
aws s3api put-bucket-versioning \  
--bucket DOC-EXAMPLE-DESTINATION-BUCKET \  
--versioning-configuration Status=Enabled \  
--profile acctA
```

4. Als Nächstes erstellen Sie eine IAM Servicerolle. Sie geben diese Rolle in der Replikationskonfiguration an, die Sie zur *DOC-EXAMPLE-SOURCE-BUCKET* Bucket später.

Amazon S3 übernimmt diese Rolle, um Objekte in Ihrem Namen zu replizieren. Sie erstellen eine IAM Rolle in zwei Schritten:

- Erstellen Sie eine Servicerolle.
- Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

a. Gehen Sie wie folgt vor, um eine IAM Servicerolle zu erstellen:

- i. Kopieren Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-trust-policy-kmsobj.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie weist dem Amazon-S3-Service-Prinzipal die Berechtigungen zum Annehmen der Rolle zu, damit Amazon S3 Aufgaben in Ihrem Namen durchführen kann.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- ii. Verwenden Sie den folgenden Befehl, um die Rolle zu erstellen:

```
$ aws iam create-role \
--role-name replicationRolekmsobj \
--assume-role-policy-document file://s3-role-trust-policy-kmsobj.json \
--profile acctA
```

b. Als Nächstes fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu. Diese Zugriffsrichtlinie erteilt Berechtigungen für verschiedene Amazon-S3-Bucket- und -Objektaktionen.

- i. Kopieren Sie die folgende Berechtigungsrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-permissions-policykmsobj.json` im aktuellen

Verzeichnis auf Ihrem lokalen Computer. Sie werden eine IAM Rolle erstellen und ihr die Richtlinie später hinzufügen.

⚠ Important

In der Berechtigungsrichtlinie geben Sie Folgendes an AWS KMS SchlüsselIDs, der für die Verschlüsselung der *amzn-s3-demo-destination-bucket* Buckets *amzn-s3-demo-source-bucket* und verwendet wird. Sie müssen zwei separate KMS Schlüssel für und erstellen *amzn-s3-demo-source-bucket* *amzn-s3-demo-destination-bucket* Eimer. AWS KMS keys werden nicht außerhalb der geteilt AWS-Region in dem sie erstellt wurden.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Action":[
        "s3:ListBucket",
        "s3:GetReplicationConfiguration",
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Effect":"Allow",
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-source-bucket",
        "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
      ]
    },
    {
      "Action":[
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
      ],
      "Effect":"Allow",
      "Condition":{"
        "StringLikeIfExists":{"
          "s3:x-amz-server-side-encryption":["
            "aws:kms",
```

```

        "AES256",
        "aws:kms:dsse"
    ],
    "s3:x-amz-server-side-encryption-aws-kms-key-id":[
        "AWS KMS key IDs(in ARN format) to use for encrypting
object replicas"
    ]
    }
},
"Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
},
{
    "Action":[
        "kms:Decrypt"
    ],
    "Effect":"Allow",
    "Condition":{"
        "StringLike":{"
            "kms:ViaService":"s3.us-east-1.amazonaws.com",
            "kms:EncryptionContext:aws:s3:arn":[
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
            ]
        }
    }
},
    "Resource":[
        "AWS KMS key IDs(in ARN format) used to encrypt source
objects."
    ]
},
{
    "Action":[
        "kms:Encrypt"
    ],
    "Effect":"Allow",
    "Condition":{"
        "StringLike":{"
            "kms:ViaService":"s3.us-west-2.amazonaws.com",
            "kms:EncryptionContext:aws:s3:arn":[
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        }
    }
},
    "Resource":[

```

```

        "AWS KMS key IDs(in ARN format) to use for encrypting object
        replicas"
      ]
    }
  ]
}

```

- ii. Erstellen Sie eine Richtlinie und hängen Sie sie an die Rolle an.

```

$ aws iam put-role-policy \
--role-name replicationRolekmsobj \
--policy-document file:///s3-role-permissions-policykmsobj.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA

```

5. Als Nächstes fügen Sie die folgende Replikationskonfiguration zum *amzn-s3-demo-source-bucket*-Bucket hinzu. Sie weist Amazon S3 an, Objekte mit dem Präfix *Tax/* in den *amzn-s3-demo-destination-bucket*-Bucket zu replizieren.

Important

In der Replikationskonfiguration geben Sie die IAM Rolle an, die Amazon S3 übernehmen kann. Dies können Sie nur tun, wenn Sie über die `iam:PassRole`-Berechtigung verfügen. Das Profil, das Sie im CLI Befehl angeben, muss über diese Berechtigung verfügen. Weitere Informationen finden Sie unter [Einem Benutzer Berechtigungen zur Übergabe einer Rolle an einen AWS-Service](#) im IAM-Benutzerhandbuch.

```

<ReplicationConfiguration>
  <Role>IAM-RoLe-ARN</Role>
  <Rule>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <SourceSelectionCriteria>

```

```

    <SseKmsEncryptedObjects>
      <Status>Enabled</Status>
    </SseKmsEncryptedObjects>
  </SourceSelectionCriteria>
  <Destination>
    <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    <EncryptionConfiguration>
      <ReplicaKmsKeyID>AWS KMS key IDs to use for encrypting object replicas</
ReplicaKmsKeyID>
    </EncryptionConfiguration>
  </Destination>
</Rule>
</ReplicationConfiguration>

```

Um eine Replikationskonfiguration zum *amzn-s3-demo-source-bucket*-Bucket hinzuzufügen, gehen Sie wie folgt vor:

- a. Das Tool AWS CLI erfordert, dass Sie die Replikationskonfiguration als angebenJSON. Speichern Sie Folgendes JSON in einer Datei (`replication.json`) im aktuellen Verzeichnis auf Ihrem lokalen Computer.

```

{
  "Role": "IAM-RoLe-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": {
        "Status": "Disabled"
      },
      "Filter": {
        "Prefix": "Tax"
      },
      "Destination": {
        "Bucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket",
        "EncryptionConfiguration": {
          "ReplicaKmsKeyID": "AWS KMS key IDs (in ARN format) to use for
encrypting object replicas"
        }
      },
      "SourceSelectionCriteria": {
        "SseKmsEncryptedObjects": {

```

```
    "Status": "Enabled"
  }
}
]
```

- b. Bearbeiten Sie denJSON, um Werte für den *amzn-s3-demo-destination-bucket* Bucket *AWS KMS key IDs (in ARN format)*, und anzugeben *IAM-role-ARN*. Speichern Sie die Änderungen.
- c. Verwenden Sie den folgenden Befehl, um die Replikationskonfiguration zu Ihrem *amzn-s3-demo-source-bucket*-Bucket hinzuzufügen. Stellen Sie sicher, dass Sie den Namen für den *amzn-s3-demo-source-bucket*-Bucket angeben.

```
$ aws s3api put-bucket-replication \  
--replication-configuration file://replication.json \  
--bucket amzn-s3-demo-source-bucket \  
--profile acctA
```

6. Testen Sie die Konfiguration, um zu überprüfen, ob die verschlüsselten Objekte repliziert werden. Führen Sie in der Amazon-S3-Konsole die folgenden Schritte aus:
 - a. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
 - b. Erstellen Sie im *amzn-s3-demo-source-bucket*-Bucket einen Ordner mit dem Namen Tax.
 - c. Fügen Sie Beispielobjekte zum Ordner hinzu. Achten Sie darauf, die Verschlüsselungsoption auszuwählen und Ihren KMS Schlüssel zur Verschlüsselung der Objekte anzugeben.
 - d. Stellen Sie sicher, dass der *amzn-s3-demo-destination-bucket* Bucket die Objektreplicate enthält und dass sie mit dem KMS Schlüssel verschlüsselt sind, den Sie in der Konfiguration angegeben haben. Weitere Informationen finden Sie unter [the section called "Abrufen des Replikationsstatus"](#).

Verwendung der AWS SDKs

Ein Code-Beispiel, das zeigt, wie eine Replikationskonfiguration hinzugefügt wird, finden Sie unter [Verwendung der AWS SDKs](#). Sie müssen die Replikationskonfiguration entsprechend ändern.

Weitere konzeptuelle Informationen finden Sie unter [Replikation verschlüsselter Objekte \(SSE-C, SSE -S3, -, SSE -KMS\) DSSE KMS](#).

Replizieren von Metadatenänderungen mit der Synchronisierung von Amazon-S3-Replikatänderungen

Die Amazon S3 S3-Replikatmodifikationssynchronisierung kann Ihnen dabei helfen, ACLs Objektmetadaten wie Tags und Object Lock-Einstellungen zwischen Replikaten und Quellobjekten replizieren zu lassen. Standardmäßig repliziert Amazon S3 Metadaten aus den Quellobjekten nur auf die Replikate. Wenn die Synchronisierung von Replikatänderungen aktiviert ist, repliziert Amazon S3 Metadatenänderungen an den Replikatkopien zurück auf das Quellobjekt, sodass die Replikation bidirektional wird.

Aktivieren der Synchronisierung von Replikatänderungen

Sie können die Synchronisierung von Amazon-S3-Replikatänderungen mit neuen oder vorhandenen Replikationsregeln verwenden. Sie können sie auf einen gesamten S3-Bucket oder auf Amazon-S3-Objekte anwenden, die ein bestimmtes Präfix haben.

Informationen zum Aktivieren der Synchronisierung von Replikatänderungen über die Amazon-S3-Konsole finden Sie unter [Beispiele für die Konfiguration der Live-Replikation](#). Dieses Thema enthält Anweisungen zur Aktivierung der Synchronisation von Replikatänderungen in Ihrer Replikationskonfiguration, wenn Buckets denselben oder unterschiedlichen Buckets gehören AWS-Konten.

Um die Synchronisierung von Replikatänderungen zu aktivieren, verwenden Sie AWS Command Line Interface (AWS CLI), müssen Sie dem Bucket, der die Replikate enthält, eine Replikationskonfiguration hinzufügen, wenn die Option aktiviert ist. `ReplicaModifications` Um die bidirektionale Replikation einzurichten, erstellen Sie eine Replikationsregel aus dem Quell-Bucket (*amzn-s3-demo-bucket1*) zu dem Bucket, der die Replikate enthält (*amzn-s3-demo-bucket2*). Erstellen Sie dann eine zweite Replikationsregel aus dem Bucket, der die Replikate enthält (*amzn-s3-demo-bucket2*) zum Quell-Bucket (*amzn-s3-demo-bucket1*). Eimer können sich im selben oder in verschiedenen befinden, AWS-Regionen.

Note

Sie müssen die Replikatänderungssynchronisierung für beide Buckets aktivieren, um Änderungen an den Replikatmetadaten wie Objektzugriffskontrolllisten (ACLs), Objekttags oder Objektsperreinstellungen für die replizierten Objekte zu replizieren. Wie alle Replikationsregeln können diese Regeln entweder auf den gesamten Amazon-S3-Bucket

oder auf eine nach Präfix oder Objekt-Tags gefilterte Teilmenge von Amazon-S3-Objekten angewendet werden.

In der folgenden Beispielkonfiguration repliziert Amazon S3 Metadatenänderungen unter dem Präfix *Tax* zum Bucket *amzn-s3-demo-bucket*, das die Quellobjekte enthalten würde.

```
{
  "Rules": [
    {
      "Status": "Enabled",
      "Filter": {
        "Prefix": "Tax"
      },
      "SourceSelectionCriteria": {
        "ReplicaModifications": {
          "Status": "Enabled"
        }
      },
      "Destination": {
        "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
      },
      "Priority": 1
    }
  ],
  "Role": "IAM-Role-ARN"
}
```

Vollständige Anweisungen zum Erstellen von Replikationsregeln finden Sie unter AWS CLI, finden Sie unter [Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos](#).

Replizieren von Löschmarkierungen auf Buckets

Wenn S3-Replikation aktiviert ist und ein Objekt im Quell-Bucket gelöscht wird, fügt Amazon S3 standardmäßig nur im Quell-Bucket eine Löschmarkierung hinzu. Dies schützt Daten vor missbräuchlichen Löschungen.

Wenn Sie die Löschmarkierungsreplikation aktiviert haben, werden diese Markierungen in die Ziel-Buckets kopiert und Amazon S3 verhält sich so, als sei das Objekt sowohl in den Quell- als auch in den Ziel-Buckets gelöscht worden. Weitere Informationen zur Funktionsweise von Löschmarkierungen finden Sie unter [Arbeiten mit Löschmarkierungen](#).

Note

Die Replikation von Löschmarkierungen wird für Tag-basierte Replikationsregeln nicht unterstützt. Bei der Replikation mit Löschmarkern werden auch nicht die 15 Minuten eingehalten, die bei der Verwendung von S3 Replication Time Control SLA gewährt werden.

Wenn Sie nicht die neueste Version der Replikations-Konfiguration verwenden, wirken sich Löschvorgänge unterschiedlich auf die Replikation aus. Weitere Informationen finden Sie unter [Auswirkungen von Löschvorgängen auf die Replikation](#).

Aktivieren der Löschmarkierungs-Replikation

Sie können die Löschmarkierungs-Replikation mit einer neuen oder einer vorhandenen Replikationsregel verwenden. Sie können sie auf einen gesamten S3-Bucket oder auf Amazon-S3-Objekte anwenden, die ein bestimmtes Präfix haben.

Note

Wenn Sie die Replikation von Löschmarkern aktivieren und Ihr Bucket über eine S3-Lebenszyklusablaufregel verfügt, werden die durch die S3-Lebenszyklusablaufregel hinzugefügten Löschmarkierungen nicht in den Ziel-Bucket repliziert.

Informationen zum Aktivieren der Löschung von Markierungsreplikationen mit der Amazon-S3-Konsole finden Sie unter [Verwenden der S3-Konsole](#). Dieses Thema enthält Anweisungen zur Aktivierung der Replikation mit Löschmarkern in Ihrer Replikationskonfiguration, wenn Buckets demselben oder unterschiedlichen Buckets gehören AWS-Konten.

Um die Replikation mit Löschmarkern zu aktivieren, verwenden Sie AWS Command Line Interface (AWS CLI), müssen Sie dem Quell-Bucket eine Replikationskonfiguration mit `DeleteMarkerReplication` aktivierter Option hinzufügen.

In der folgenden Beispielkonfiguration werden Löschmarkierungen in den Ziel-Bucket repliziert **DOC-EXAMPLE-BUCKET** für Objekte unter dem Präfix **Tax**.

```
{
  "Rules": [
    {
      "Status": "Enabled",
```

```
    "Filter": {
      "Prefix": "Tax"
    },
    "DeleteMarkerReplication": {
      "Status": "Enabled"
    },
    "Destination": {
      "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    },
    "Priority": 1
  }
],
"Role": "IAM-Role-ARN"
}
```

Vollständige Anweisungen zum Erstellen von Replikationsregeln finden Sie unter AWS CLI, siehe [Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos](#) im Abschnitt Exemplarische Vorgehensweisen zur Replikation.

Live-Replikation verwalten oder pausieren

Live-Replikation ist das automatische, asynchrone Kopieren von Objekten über Buckets hinweg, die sich im selben oder in unterschiedlichen Buckets befinden AWS-Regionen. Nachdem Sie Ihre Replikationskonfiguration eingerichtet haben, repliziert Amazon S3 neu erstellte Objekte und Objektaktualisierungen von einem Quell-Bucket in einen oder mehrere angegebene Ziel-Buckets.


Sie verwenden die Amazon-S3-Konsole, um dem Quell-Bucket Replikationsregeln hinzuzufügen. Replikationsregeln definieren, welche Quell-Bucket-Objekte repliziert werden sollen, und den Ziel-Bucket/die Ziel-Buckets, in dem/denen die replizierten Objekte gespeichert werden sollen. Weitere Informationen zur Replikation finden Sie unter [Überblick über das Replizieren von Objekten](#).

Sie können Replikationsregeln auf der Seite Replication (Replikation) verwalten. Sie können Replikationsregeln hinzufügen, anzeigen, aktivieren, deaktivieren oder löschen. Sie können auch die Priorität Ihrer Replikationsregeln ändern. Weitere Informationen zum Hinzufügen von Replikationsregeln zu einem Bucket finden Sie unter [Verwenden der S3-Konsole](#).

Um die Replikationsregeln für einen S3-Bucket mithilfe der Amazon S3 S3-Konsole zu verwalten

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.

3. Wählen Sie auf der Registerkarte Allgemeine Buckets den Namen des gewünschten Buckets aus.
4. Wählen Sie die Registerkarte Verwaltung und scrollen Sie dann nach unten zu Replikationsregeln.
5. Sie können Ihre Replikationsregeln auf folgende Weise ändern:
 - Um eine Replikationsregel zu aktivieren oder zu deaktivieren, wählen Sie das Optionsfeld links neben der Regel. Wählen Sie im Menü Aktionen die Option Regel aktivieren oder Regel deaktivieren aus. Sie können auch alle Regeln im Bucket im Menü Aktionen deaktivieren, aktivieren oder löschen.

 Note

Wenn Sie eine Replikationsregel deaktivieren und die Regel später wieder aktivieren, werden alle neuen oder geänderten Objekte, die während der Deaktivierung der Regel nicht repliziert wurden, nicht automatisch repliziert, wenn die Regel wieder aktiviert wird. Um diese Objekte zu replizieren, müssen Sie S3 Batch Replication verwenden. Weitere Informationen finden Sie unter [the section called "Replizieren vorhandener Objekte"](#).

- Um die Priorität einer Regel zu ändern, wählen Sie das Optionsfeld links neben der Regel und wählen Sie dann Regel bearbeiten aus.

Regelprioritäten werden festgelegt, um Konflikte zu vermeiden, die durch Objekte verursacht werden, die mehreren Regeln unterliegen. Wenn sich Regeln überschneiden, verwendet Amazon S3 die Regelpriorität, um die Regel zu ermitteln, die angewendet werden muss. Je höher die Zahl, desto höher die Priorität. Weitere Informationen zur Priorität von Regeln finden Sie unter [Replikationskonfiguration](#).

Die Replikation wird angehalten oder beendet

Um die Replikation vorübergehend anzuhalten und sie später automatisch wieder aufzunehmen, können Sie die Aktion in `aws:s3:bucket-pause-replication` AWS Fault Injection Service. Weitere Informationen finden Sie unter [aws:s3:bucket-pause-replication](#) und [Unterbrechen Sie die S3-Replikation](#) im AWS Fault Injection Service Benutzerleitfaden.

Um die Replikation in Amazon S3 zu beenden, empfehlen wir, Ihre Replikationsregeln zu deaktivieren. Wenn Sie eine Replikationsregel deaktivieren und die Regel später wieder aktivieren,

werden alle neuen oder geänderten Objekte, die nicht repliziert wurden, während die Regel deaktiviert war, nicht automatisch repliziert, wenn die Regel erneut aktiviert wird. Um diese Objekte zu replizieren, müssen Sie S3 Batch Replication verwenden. Weitere Informationen finden Sie unter [the section called “Replizieren vorhandener Objekte”](#).

Die Replikation wird auch beendet, wenn Sie das entfernen AWS Identity and Access Management (IAM) Rolle, AWS Key Management Service (AWS KMS) Berechtigungen oder die Bucket-Policy-Berechtigungen, die Amazon S3 die erforderlichen Berechtigungen gewähren. Wir empfehlen diese Ansätze jedoch nicht, da sie dazu führen, dass die Replikation fehlschlägt. Amazon S3 meldet den Replikationsstatus für betroffene Objekte als FAILED. Wenn Berechtigungen später wiederhergestellt werden, werden Objekte, die als markiert FAILED sind, nicht automatisch repliziert. Um diese Objekte zu replizieren, müssen Sie S3 Batch Replication verwenden.

Replizieren bestehender Objekte mit S3-Batch-Replikation

Mithilfe von S3 Batch Replication können Sie die folgenden Objekttypen replizieren:

- Objekte, die vor der Einrichtung einer Replikationskonfiguration existierten
- Objekte, die zuvor repliziert wurden
- Objekte, deren Replikation fehlgeschlagen ist

Sie können diese Objekte bei Bedarf replizieren, indem Sie einen Batch Operations-Job verwenden. Die S3-Batch-Replikation unterscheidet sich von der Live-Replikation, bei der kontinuierlich und automatisch neue Objekte in Amazon S3 S3-Buckets repliziert werden.

Um mit Batch Replication zu beginnen, können Sie:

- Batch-Replikation für eine neue Replikationsregel oder ein neues Ziel initiieren — Sie können einen einmaligen Batch-Replikationsauftrag erstellen, wenn Sie die erste Regel in einer neuen Replikationskonfiguration erstellen oder wenn Sie einer bestehenden Konfiguration über die Amazon S3 S3-Konsole ein neues Ziel hinzufügen.
- Batch-Replikation für eine bestehende Replikationskonfiguration initiieren — Sie können einen neuen Batch-Replikationsauftrag erstellen, indem Sie S3 Batch Operations über die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI), AWS SDKs, oder der Amazon S3 RESTAPI.

Wenn der Batch-Replikationsauftrag abgeschlossen ist, erhalten Sie einen Abschlussbericht. Weitere Informationen dazu, wie Sie den Bericht verwenden können, um den Auftrag zu untersuchen, finden Sie unter [Verfolgen von Auftragsstatus- und Abschluss](#).

Überlegungen zur S3-Batch-Replikation

- Ihr Quell-Bucket muss über eine vorhandene Replikationskonfiguration verfügen. Informationen zum Aktivieren der Replikation finden Sie unter [Live-Replikation einrichten](#) und [Beispiele für die Konfiguration der Live-Replikation](#).
- Wenn Sie S3 Lifecycle für Ihren Bucket konfiguriert haben, empfehlen wir, Ihre Lebenszyklusregeln zu deaktivieren, solange der Batch Replication-Job aktiv ist. Auf diese Weise wird die Parität zwischen den Quell- und Ziel-Buckets gewährleistet. Andernfalls könnten diese Buckets voneinander abweichen, und der Ziel-Bucket ist kein exaktes Replikat des Quell-Buckets. Betrachten wir folgendes Beispielszenario:
 - Ihr Quell-Bucket enthält mehrere Versionen eines Objekts und eine Löschkennzeichnung für dieses Objekt.
 - Ihre Quell- und Ziel-Buckets verfügen über eine Lebenszyklus-Konfiguration zum Entfernen abgelaufener Löschkennzeichnungen.

In diesem Szenario repliziert die Batch-Replikation möglicherweise die Löschkennzeichnung in den Ziel-Bucket, bevor die Objektversionen repliziert werden. Dieses Verhalten könnte dazu führen, dass Ihre Lebenszykluskonfiguration die Löschkennzeichnung als abgelaufen markiert und die Löschkennzeichnung aus dem Ziel-Bucket entfernt, bevor die Objektversionen repliziert werden.

- Das Tool AWS Identity and Access Management (IAM) Die Rolle, die Sie für die Ausführung des Batch Operations-Jobs angeben, muss über die erforderlichen Berechtigungen verfügen, um den zugrunde liegenden Batch-Replikationsvorgang auszuführen. Weitere Informationen zum Erstellen von IAM Rollen finden Sie unter [Konfiguration von IAM Richtlinien für die Batch-Replikation](#).
- Für die Batch-Replikation ist ein Manifest erforderlich, das von Amazon S3 generiert werden kann. Das generierte Manifest muss in derselben gespeichert werden AWS-Region wie der Quell-Bucket. Wenn Sie das Manifest nicht generieren möchten, können Sie einen Amazon S3 S3-Inventarbericht oder eine CSV Datei bereitstellen, die die Objekte enthält, die Sie replizieren möchten.
- Die Batch-Replikation unterstützt nicht die erneute Replikation von Objekten, die mit der Versions-ID des Objekts aus dem Ziel-Bucket gelöscht wurden. Wenn Sie diese Objekte erneut replizieren möchten, können Sie die Quellobjekte mit einem Batch-Kopierauftrag kopieren. Wenn Sie diese Objekte an Ort und Stelle kopieren, werden neue Versionen der Objekte im Quell-Bucket erstellt

und die Replikation in den Ziel-Bucket automatisch initiiert. Durch Löschen und Neuerstellen des Ziel-Buckets wird die Replikation nicht initiiert.

Weitere Informationen zu Batch Copy finden Sie unter [Beispiele, die Batch-Vorgänge zum Kopieren von Objekten verwenden](#).

- Wenn Sie eine Replikationsregel für den S3-Bucket verwenden, stellen Sie sicher, dass Sie [Ihre Replikationskonfiguration aktualisieren](#), indem Sie der IAM Rolle, die der Replikationsregel zugeordnet ist, die entsprechenden Berechtigungen zum Replizieren von Objekten gewähren. Diese IAM Rolle muss über die erforderlichen Berechtigungen verfügen, um die Replikation sowohl für den Quell- als auch für den Ziel-Bucket durchzuführen.
- Wenn Sie innerhalb eines kurzen Zeitraums mehrere Batch-Replikationsaufträge für denselben Bucket einreichen, führt Amazon S3 diese Jobs gleichzeitig aus.
- Wenn Sie mehrere Batch-Replikationsaufträge für zwei verschiedene Buckets einreichen, beachten Sie, dass Amazon S3 möglicherweise nicht alle Jobs gleichzeitig ausführt. Wenn Sie die Anzahl der Batch-Replikationsaufträge, die gleichzeitig auf Ihrem Konto ausgeführt werden können, überschreiten, pausiert Amazon S3 die Jobs mit niedrigerer Priorität, um an den Jobs mit höherer Priorität zu arbeiten. Nachdem die Aufgaben mit der höheren Priorität abgeschlossen wurden, werden alle unterbrochenen Jobs wieder aktiv.
- Die Batch-Replikation wird nicht für Objekte unterstützt, die in den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive gespeichert sind.
- Um S3 Intelligent-Tiering-Objekte, die in den Speicherstufen Archive Access oder Deep Archive Access gespeichert sind, stapelweise zu replizieren, müssen Sie zunächst eine [Wiederherstellungsanforderung](#) initiieren und warten, bis die Objekte auf die Stufe für häufigen Zugriff verschoben wurden.

Angeben eines Manifests für einen Batch-Replikationsauftrag

Ein Manifest ist ein Amazon-S3-Objekt, das Objektschlüssel enthält, die Amazon S3 bearbeiten soll. Wenn Sie einen Batch-Replikationsauftrag erstellen möchten, müssen Sie entweder ein benutzergeneriertes Manifest angeben oder Amazon S3 ein Manifest auf der Grundlage Ihrer Replikationskonfiguration generieren lassen.

Wenn Sie ein benutzergeneriertes Manifest bereitstellen, muss es in Form eines Amazon S3 S3-Lagerberichts oder einer CSV Datei vorliegen. Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, müssen Sie die Version IDs für die Objekte angeben. Nur das Objekt

mit der im Manifest angegebenen Versions-ID wird repliziert. Weitere Informationen zum Angeben eines Manifests finden Sie unter [Angeben eines Manifests](#).

Wenn Sie sich dafür entscheiden, dass Amazon S3 in Ihrem Namen eine Manifestdatei generiert, verwenden die aufgelisteten Objekte denselben Quell-Bucket, dasselbe Präfix und dieselben Tags wie alle Ihre Replikationskonfigurationen des Quell-Buckets. Mit einem generierten Manifest repliziert Amazon S3 alle geeigneten Versionen Ihrer Objekte.

Note

Wenn Sie Amazon S3 das Manifest generieren lassen möchten, muss das Manifest im selben Verzeichnis gespeichert werden. AWS-Region als Quell-Bucket.

Filter für einen Batch-Replikationsauftrag

Bei der Erstellung Ihres Batch-Replikationsjobs können Sie optional zusätzliche Filter angeben, z. B. das Datum der Objekterstellung und den Replikationsstatus, um den Umfang des Jobs zu reduzieren.

Sie können Objekte filtern, die repliziert werden sollen, basierend auf dem `ObjectReplicationStatuses`-Wert, indem Sie einen oder mehrere der folgenden Werte angeben:

- "NONE" – Zeigt an, dass Amazon S3 noch nie versucht hat, das Objekt zu replizieren.
- "FAILED"— Zeigt an, dass Amazon S3 versucht hat, das Objekt zuvor zu replizieren, dies jedoch fehlgeschlagen ist.
- "COMPLETED" – Zeigt an, dass Amazon S3 das Objekt zuvor erfolgreich repliziert hat.
- "REPLICA"— Zeigt an, dass es sich um ein Replikat-Objekt handelt, das Amazon S3 aus einer anderen Quelle repliziert hat.

Weitere Informationen zum Replikationsstatus finden Sie unter [Abrufen von Replikationsstatusinformationen](#).

Wenn Sie Ihren Batch-Replikationsauftrag nicht filtern, versucht Batch Operations, alle Objekte (unabhängig von ihrem `ObjectReplicationStatus`) in Ihrem Manifest zu replizieren, die den Regeln in Ihrer Replikationskonfiguration entsprechen, mit Ausnahme bestimmter Objekte, die standardmäßig nicht repliziert werden. Weitere Informationen finden Sie unter [the section called "Was wird mit Replikationskonfigurationen nicht repliziert?"](#)

Abhängig von Ihrem Ziel können Sie einen oder mehrere der folgenden Werte festlegen `ObjectReplicationStatuses`:

- Um nur vorhandene Objekte zu replizieren, die noch nie repliziert wurden, schließen Sie nur ein. "NONE"
- Um erneut zu versuchen, nur Objekte zu replizieren, die zuvor nicht repliziert werden konnten, schließen Sie nur ein. "FAILED"
- Um sowohl vorhandene Objekte zu replizieren als auch erneut zu versuchen, Objekte zu replizieren, die zuvor nicht repliziert werden konnten, schließen Sie sowohl als auch ein. "NONE" "FAILED"
- Um einen Ziel-Bucket mit Objekten aufzufüllen, die an ein anderes Ziel repliziert wurden, schließen Sie Folgendes ein: "COMPLETED"
- Um Objekte zu replizieren, die zuvor repliziert wurden, schließen Sie ein. "REPLICA"

Abschlussbericht zur Batchreplikation

Wenn Sie einen Batch-Replikationsauftrag erstellen, können Sie einen CSV Abschlussbericht anfordern. In diesem Bericht werden Objekte, Erfolgs- oder Fehlercodes, Ausgaben und Beschreibungen der Replikation angezeigt. Weitere Informationen zur Auftragsverfolgung und zu Abschlussberichten finden Sie unter [Abschlussberichte](#).

Eine Liste der Fehlercodes und Beschreibungen der Replikationsfehler finden Sie unter [Gründe für das Fehlschlagen der Replikation in Amazon S3](#).

Informationen zur Problembehandlung bei der Batch-Replikation finden Sie unter [Fehler bei der Batchreplikation](#).

Erste Schritte mit der Batch-Replikation

Weitere Informationen zur Verwendung der Batch-Replikation finden Sie unter [Tutorial: Replizieren vorhandener Objekte in Ihren Amazon S3-Buckets mit S3 Batch Replication](#).

Konfiguration von IAM Richtlinien für die Batch-Replikation

Da es sich bei S3 Batch Replication um eine Art Batch-Operations-Job handelt, müssen Sie einen Batch erstellen. AWS Identity and Access Management (IAM) Rolle, um Amazon S3 S3-Berechtigungen zur Durchführung von Aktionen in Ihrem Namen zu erteilen. Sie müssen der IAM Rolle Batch Operations auch eine IAM Batch-Replikationsrichtlinie hinzufügen. Im folgenden

Beispiel wird eine IAM Rolle erstellt, die Batch Operations die Berechtigung erteilt, einen Batch-Replikationsauftrag zu initiieren.

IAMRolle und Richtlinie erstellen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie unter Access management (Zugriffsverwaltung) Roles (Rollen) aus.
3. Wählen Sie Create Role aus.
4. Wählen Sie aus.AWS-Serviceals Typ der vertrauenswürdigen Entität, Amazon S3 als Service und S3 Batch Operations als Anwendungsfall.
5. Wählen Sie Next: Permissions aus.
6. Wählen Sie Create Policy (Richtlinie erstellen) aus.
7. Wählen Sie basierend auf Ihrem Manifest eine der folgenden Richtlinien aus JSONund fügen Sie sie ein.

Note

Wenn Sie ein Manifest erstellen oder ein Manifest liefern, sind andere Berechtigungen erforderlich. Weitere Informationen finden Sie unter [Angeben eines Manifests für einen Batch-Replikationsauftrag](#).

Richtlinie für die Verwendung und Speicherung eines von S3 generierten Manifests

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:InitiateReplication"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::*** replication source bucket ***/*"
      ]
    }
  ],
  {
```

```

    "Action":[
      "s3:GetReplicationConfiguration",
      "s3:PutInventoryConfiguration"
    ],
    "Effect":"Allow",
    "Resource":[
      "arn:aws:s3:::*** replication source bucket ***"
    ]
  },
  {
    "Action":[
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Effect":"Allow",
    "Resource":[
      "arn:aws:s3:::*** manifest bucket ***/*"
    ]
  },
  {
    "Effect":"Allow",
    "Action":[
      "s3:PutObject"
    ],
    "Resource":[
      "arn:aws:s3:::*** completion report bucket ****/*",
      "arn:aws:s3:::*** manifest bucket ****/*"
    ]
  }
]
}

```

Richtlinie bei Verwendung eines vom Benutzer bereitgestellten Manifests

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Action":[
        "s3:InitiateReplication"
      ],
      "Effect":"Allow",
      "Resource":[

```

```

        "arn:aws:s3:::*** replication source bucket ***/*"
    ],
    },
    {
        "Action": [
            "s3:GetObject",
            "s3:GetObjectVersion"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::*** manifest bucket ***/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::*** completion report bucket ****/*"
        ]
    }
]
}

```

8. Wählen Sie Next: Tags (Weiter: Tags) aus.
9. Klicken Sie auf Weiter: Prüfen.
10. Wählen Sie einen Namen für die Richtlinie und wählen Sie dann Create policy (Richtlinie erstellen) aus.
11. Fügen Sie diese Richtlinie an Ihre Rolle an und wählen Sie Next: Tags (Weiter: Tags).
12. Wählen Sie Weiter: Prüfen aus.
13. Wählen Sie einen Namen für die Rolle und wählen Sie Create role (Rolle erstellen).

Überprüfen der Vertrauensrichtlinie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie unter Access management (Zugriffsverwaltung) Roles (Rollen) und Ihre neu erstellte Rolle aus.

3. Wählen Sie auf der Registerkarte Trust relationships (Vertrauensstellungen) die Option Edit trust relationship (Vertrauensstellung bearbeiten).
4. Stellen Sie sicher, dass diese Rolle die folgende Vertrauensrichtlinie verwendet:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batchoperations.s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Erstellen eines Batch-Replikationsauftrags für eine erste Replikationsregel oder ein neues Ziel


Wenn Sie die erste Regel in einer neuen Replikationskonfiguration erstellen oder einer vorhandenen Konfiguration ein neues Ziel hinzufügen, verwenden Sie den AWS Management Console, können Sie optional einen Batch-Replikationsauftrag erstellen.

Informationen zum Verwenden der Batch-Replikation für eine vorhandene Konfiguration ohne Hinzufügen eines neuen Ziels finden Sie unter [Erstellen eines Batch-Replikationsauftrags für vorhandene Replikationsregeln](#).

Verwenden der Batch-Replikation für eine neue Replikationsregel oder ein neues Replikationsziel über AWS Management Console

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste von Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie replizieren möchten.
3. Wählen Sie zum Erstellen einer neuen Replikationsregel oder zum Bearbeiten einer vorhandenen Regel Management (Verwaltung) und scrollen Sie nach unten zu Replication rules (Replikationsregeln):

- Wählen Sie zum Erstellen einer neuen Replikationsregel Create replication rule (Erstellen einer Replikationsregel).

 Note

Beispiele zum Einrichten einer grundlegenden Replikationsregel finden Sie unter: [Beispiele für die Konfiguration der Live-Replikation](#).

- Wählen Sie zum Bearbeiten einer vorhandenen Replikationsregel die entsprechende Regel aus und wählen Sie dann Edit rule (Regel bearbeiten).
4. Erstellen Sie Ihre neue Replikationsregel oder bearbeiten Sie das Ziel für Ihre vorhandene Replikationsregel und wählen Sie Save (Speichern).

Nachdem Sie die erste Regel in einer neuen Replikationskonfiguration erstellt oder eine vorhandene Konfiguration bearbeitet haben, um ein neues Ziel hinzuzufügen, wird der Dialog Replicate existing objects? (Vorhandene Objekte replizieren?) angezeigt, über den Sie einen Batch-Replikationsauftrag erstellen können.

5. Wenn Sie diesen Auftrag jetzt ausführen möchten, wählen Sie Ja, vorhandene Objekte replizieren aus.

Wenn Sie diesen Auftrag zu einem späteren Zeitpunkt ausführen möchten, wählen Sie Nein, keine vorhandenen Objekte replizieren aus.

6. Erstellen Sie Ihren S3-Batch-Replikationsauftrag. Der S3-Batch-Replikationsauftrag hat mehrere Einstellungen:

Option zur Auftragsausführung

Wenn der S3-Batch-Replikationsauftrag sofort ausgeführt werden soll, können Sie Job runs automatically when ready (Auftrag wird automatisch ausgeführt, wenn er bereit ist) wählen. Wenn Sie den Auftrag zu einem späteren Zeitpunkt ausführen möchten, wählen Sie Job waits to be run when ready (Auftrag wartet auf Ausführung, wenn er bereit ist).

Wenn Sie Job runs automatically when ready (Auftrag wird automatisch ausgeführt, wenn er bereit ist) wählen, können Sie kein Batchvorgangs-Manifest erstellen und speichern. Um das Batchvorgangs-Manifest zu speichern, wählen Sie Job waits to be run when ready (Auftrag wartet auf Ausführung, wenn er bereit ist) aus.

Batchvorgangs-Manifest

Das Manifest ist eine Liste aller Objekte, für die die festgelegte Aktion ausgeführt werden soll. Sie können das Batchvorgangs-Manifest speichern. Ähnlich wie bei S3-Inventardateien wird das Manifest als CSV Datei gespeichert und in einem Bucket gespeichert. Weitere Informationen über die Batchvorgangs-Manifeste finden Sie unter [Angeben eines Manifests](#).

Abschlussbericht

S3-Batchvorgänge führen eine Aufgabe für jedes Objekt aus, das im Manifest angegeben ist. Abschlussberichte bieten eine einfache Möglichkeit, die Ergebnisse von Aufgaben in einem konsolidierten Format ohne zusätzliche Einrichtung anzuzeigen. Sie können einen Abschlussbericht für alle oder nur für die fehlgeschlagenen Aufgaben anfordern. Weitere Informationen zu Abschlussberichten finden Sie unter [Abschlussberichte](#).

Berechtigungen

Eine der häufigsten Ursachen für Replikationsfehler sind unzureichende Berechtigungen in den bereitgestellten AWS Identity and Access Management (IAM) Rolle. Weitere Informationen zum Erstellen dieser Rolle finden Sie unter [Konfiguration von IAM Richtlinien für die Batch-Replikation](#).

7. Wählen Sie Create Batch Operations Job (Batchvorgangs-Auftrag erstellen).

Erstellen eines Batch-Replikationsauftrags für vorhandene Replikationsregeln

Sie können S3 Batch Replication für eine bestehende Replikationskonfiguration konfigurieren, indem Sie den AWS SDKs, AWS Command Line Interface (AWS CLI) oder die Amazon S3 S3-Konsole. Eine Übersicht über die Batch-Replikation finden Sie unter [Replizieren bestehender Objekte mit S3-Batch-Replikation](#).

Als Voraussetzung müssen Sie eine Batch-Operation erstellen AWS Identity and Access Management (IAM) Rolle, um Amazon S3 S3-Berechtigungen zur Ausführung von Aktionen in Ihrem Namen zu erteilen, siehe [Konfiguration von IAM Richtlinien für die Batch-Replikation](#).

Wenn der Batch-Replikationsauftrag abgeschlossen ist, erhalten Sie einen Abschlussbericht. Weitere Informationen dazu, wie Sie den Bericht verwenden können, um den Auftrag zu untersuchen, finden Sie unter [Verfolgen von Auftragsstatus- und Abschluss](#).

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Klicken Sie im Navigationsbereich der Amazon-S3-Konsole auf Batch-Vorgänge (Batch-Vorgänge).
3. Wählen Sie Create job (Auftrag erstellen) aus.
4. Wählen Sie die Region aus, in der Sie Ihren Auftrag erstellen möchten.
5. Wählen Sie das Manifest format (Manifestformat). In diesem Beispiel wird gezeigt, wie ein Manifest basierend auf einer vorhandenen S3-Replikationskonfiguration erstellt wird.

Note

Das Manifest ist eine Liste aller Objekte, für die die festgelegte Aktion ausgeführt werden soll. Weitere Informationen über die Batchvorgangs-Manifeste finden Sie unter [Angeben eines Manifests](#). Wenn Sie ein Manifest vorbereitet haben, wählen Sie S3-Inventarbericht (manifest.json) oder CSV. Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, sollten Sie die Version IDs für die Objekte angeben. Weitere Informationen zum Erstellen eines Manifests finden Sie unter [Angeben eines Manifests](#).

6. Um ein Manifest basierend auf Ihrer Replikationskonfiguration zu erstellen, wählen Sie Create manifest using S3 Replication configuration (Manifest mit der S3-Replikationskonfiguration erstellen). Wählen Sie dann den Quell-Bucket Ihrer Replikationskonfiguration aus.
7. (Optional) Sie können zusätzliche Filter wie Objekt-Erstellungsdatum und Replikationsstatus einschließen. Beispiele zum Filtern nach Replikationsstatus finden Sie unter [Angeben eines Manifests für einen Batch-Replikationsauftrag](#).
8. Um ein Manifest zu speichern, wählen Sie Save Batch Operations manifest (Manifest für Batchvorgänge speichern).
 - a. Wenn Sie ein Manifest generieren und speichern möchten, müssen Sie entweder Bucket in diesem Konto oder Bucket in einem anderen Konto wählen AWS-Konto. Geben Sie den Bucket-Namen in das Textfeld ein.

Note

Das generierte Manifest muss in derselben gespeichert werden AWS-Region wie der Quell-Bucket.

- b. Wählen Sie den Verschlüsselungstyp aus.
9. (Optional) Geben Sie eine Beschreibung ein.
10. Passen Sie die Priority (Priorität) des Auftrags bei Bedarf an. Höhere Nummern bedeuten eine höhere Priorität. Amazon S3 versucht, Aufträge mit höherer Priorität vor Aufträgen mit niedrigerer Priorität auszuführen. Weitere Informationen zur Auftragspriorität finden Sie unter [Zuweisen der Auftragspriorität](#).
11. (Optional) Generieren Sie einen Abschlussbericht. Wählen Sie zum generieren Generate completion report (Abschlussbericht generieren).

Wenn Sie einen Abschlussbericht erstellen möchten, müssen Sie entweder die Meldung von Failed tasks only (Nur fehlgeschlagenen Aufgaben) oder All tasks (Allen Aufgaben) auswählen und einen Ziel-Bucket für den Bericht angeben.

12. Wählen Sie eine gültige IAM Rolle aus.

Note

Weitere Informationen zum Erstellen einer IAM Rolle finden Sie unter [Konfiguration von IAM Richtlinien für die Batch-Replikation](#).

13. (Optional) Fügen Sie Auftrags-Tags zum Batch-Replikationsauftrag hinzu.
14. Wählen Sie Weiter.
15. Überprüfen Sie Ihre Auftrags-Konfiguration und wählen Sie Create job (Auftrag erstellen).

Verwendung der AWS CLI mit einem S3-Manifest

Das folgende Beispiel erstellt einen S3-Batch-Replikationsauftrag unter Verwendung eines von S3 generierten Manifests für AWS-Konto **111122223333**. In diesem Beispiel wird versucht, vorhandene Objekte und Objekte, die zuvor nicht repliziert werden konnten, zu replizieren. Informationen zum Filtern nach Replikationsstatus finden Sie unter [Angabe eines Manifests für einen Batch-Replikationsauftrag](#).

```
aws s3control create-job --account-id 111122223333 --operation
 '{"S3ReplicateObject':{}}' --report '{"Bucket":"arn:aws:s3:::***
 completion report bucket ****","Prefix":"batch-replication-report",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}'
 --manifest-generator '{"S3JobManifestGenerator": {"ExpectedBucketOwner":
 "111122223333", "SourceBucket": "arn:aws:s3:::*** replication source bucket
 ****", "EnableManifestOutput": false, "Filter": {"EligibleForReplication": true,
 "ObjectReplicationStatuses": ["NONE","FAILED"]}}}' --priority 1 --role-arn
 arn:aws:iam::111122223333:role/batch-Replication-IAM-policy --no-confirmation-required
 --region source-bucket-region
```

Note

Der Job muss von demselben aus initiiert werden AWS-Region Replizierungsquellen-Bucket. Die IAM Rolle `role/batch-Replication-IAM-policy` wurde zuvor erstellt. Siehe [Konfiguration von IAM Richtlinien für die Batch-Replikation](#).

Nachdem Sie einen Batch-Replikationsauftrag erfolgreich initiiert haben, erhalten Sie die Auftrags-ID als Antwort. Sie können diesen Auftrag mit dem folgenden Befehl überwachen.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-
 bucket-region
```

Verwendung der AWS CLI mit einem vom Benutzer bereitgestellten Manifest

Das folgende Beispiel erstellt einen S3-Batch-Replikationsauftrag mit einem benutzerdefinierten Manifest für AWS-Konto `111122223333`. Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, müssen Sie die Version IDs für die Objekte angeben. Es wird nur das Objekt mit der im Manifest angegebenen Versions-ID repliziert. Weitere Informationen zum Erstellen eines Manifests finden Sie unter [Angeben eines Manifests](#).

```
aws s3control create-job --account-id 111122223333 --operation
 '{"S3ReplicateObject':{}}' --report '{"Bucket":"arn:aws:s3:::***
 completion report bucket ****","Prefix":"batch-replication-report",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}'
 --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
 ["Bucket","Key","VersionId"]},"Location":{"ObjectArn":"arn:aws:s3:::*** completion
 report bucket ****/manifest.csv","ETag":"Manifest Etag"}}' --priority 1 --role-arn
```

```
arn:aws:iam::111122223333:role/batch-Replication-IAM-policy --no-confirmation-required  
--region source-bucket-region
```

Note

Der Job muss von demselben aus initiiert werden AWS-Region Replizierungsquellen-Bucket. Die IAM Rolle `role/batch-Replication-IAM-policy` wurde zuvor erstellt. Siehe [Konfiguration von IAM Richtlinien für die Batch-Replikation](#).

Nachdem Sie einen Batch-Replikationsauftrag erfolgreich initiiert haben, erhalten Sie die Auftrags-ID als Antwort. Sie können diesen Auftrag mit dem folgenden Befehl überwachen.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-  
bucket-region
```

Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status

Sie können Ihre Live-Replikationskonfigurationen und Ihre S3-Batch-Replikationsaufträge mithilfe der folgenden Mechanismen überwachen:

- S3-Replikationsmetriken — Wenn Sie S3-Replikationsmetriken aktivieren, CloudWatch gibt Amazon Metriken aus, mit denen Sie ausstehende Bytes, ausstehende Operationen und Replikationslatenz auf der Ebene der Replikationsregeln verfolgen können. Sie können S3-Replikationsmetriken über die Amazon S3 S3-Konsole und die CloudWatch Amazon-Konsole anzeigen. In der Amazon S3 S3-Konsole können Sie diese Metriken auf der Registerkarte Metriken des Quell-Buckets einsehen. Weitere Informationen zu S3-Replikationsmetriken finden Sie unter [the section called “Verwendung von S3-Replikationsmetriken”](#).
- S3 Storage Lens-Metriken — Zusätzlich zu den S3-Replikationsmetriken können Sie die replikationsbezogenen Datenschutzmetriken verwenden, die in den S3 Storage Lens-Dashboards bereitgestellt werden. Wenn Sie beispielsweise die kostenlosen Metriken in S3 Storage Lens verwenden, können Sie sich Metriken wie die Gesamtzahl der Byte, die aus dem Quell-Bucket repliziert wurden, oder die Anzahl der replizierten Objekte aus dem Quell-Bucket anzeigen lassen.

Um Ihre allgemeine Einstellung zur Replikation zu überprüfen, können Sie erweiterte Metriken in S3 Storage Lens aktivieren. Mit den erweiterten Metriken in S3 Storage Lens können Sie sehen,

über wie viele Replikationsregeln Sie verschiedener Typen verfügen, einschließlich der Anzahl der Replikationsregeln mit einem ungültigen Replikationsziel.

Weitere Informationen zur Arbeit mit Replikationsmetriken in S3 Storage Lens finden Sie unter [the section called “Replikationsmetriken in den Dashboards von S3 Storage Lens anzeigen”](#).

- S3-Ereignisbenachrichtigungen — S3-Ereignisbenachrichtigungen können Sie auf Objektebene benachrichtigen, wenn Objekte nicht an ihr Ziel repliziert werden AWS-Region oder wenn Objekte nicht innerhalb bestimmter Schwellenwerte repliziert werden. S3-Ereignisbenachrichtigungen bieten die folgenden Replikationsereignistypen: `s3:Replication:OperationFailedReplication`, `s3:Replication:OperationMissedThresholds`, `s3:Replication:OperationReplicatedAfter` und `s3:Replication:OperationNotTracked`

Amazon S3 S3-Ereignisse sind über Amazon Simple Queue Service (AmazonSQS), Amazon Simple Notification Service (AmazonSNS) oder AWS Lambda. Weitere Informationen finden Sie unter [the section called “Erhalten von Benachrichtigungen über Replikations-Fehlerereignisse”](#).

- Werte für den Replikationsstatus — Sie können auch den Replikationsstatus Ihrer Objekte abrufen. Anhand des Replikationsstatus können Sie den aktuellen Status eines Objekts ermitteln, das repliziert wird. Der Replikationsstatus eines Quellobjekts gibt entweder PENDING, COMPLETED oder FAILED zurück. Der Replikationsstatus eines Replikats gibt REPLICIA zurück.

Sie können auch Werte für den Replikationsstatus verwenden, wenn Sie S3-Batch-Replikationsaufträge erstellen. Sie können diese Statuswerte beispielsweise verwenden, um Objekte zu replizieren, die entweder nie repliziert wurden oder deren Replikation fehlgeschlagen ist.

Weitere Informationen zum Abrufen des Replikationsstatus Ihrer Objekte finden Sie unter [the section called “Abrufen des Replikationsstatus”](#). Weitere Hinweise zur Verwendung dieser Werte mit der Batch-Replikation finden Sie unter [the section called “Filter für einen Batch-Replikationsauftrag”](#).

Themen

- [Verwendung von S3-Replikationsmetriken](#)
- [Replikationsmetriken in den Dashboards von S3 Storage Lens anzeigen](#)
- [Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse](#)
- [Abrufen von Replikationsstatusinformationen](#)

Verwendung von S3-Replikationsmetriken

S3-Replikationsmetriken enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Mit Replikationsmetriken können Sie den minute-by-minute Fortschritt überwachen, indem Sie ausstehende Bytes, ausstehende Operationen, Operationen, bei denen die Replikation fehlgeschlagen ist, und die Replikationslatenz verfolgen.

Note

- S3-Replikationsmetriken werden zum gleichen Tarif wie CloudWatch benutzerdefinierte Amazon-Metriken abgerechnet. Weitere Informationen finden Sie unter [CloudWatchAmazon-Preise](#).
- Wenn Sie S3 Replication Time Control verwenden, CloudWatch beginnt Amazon 15 Minuten, nachdem Sie S3 RTC für die jeweilige Replikationsregel aktiviert haben, mit der Meldung von Replikationsmetriken.

S3-Replikationsmetriken werden automatisch aktiviert, wenn Sie S3 Replication Time Control (S3RTC) aktivieren. Sie können S3-Replikationsmetriken auch unabhängig von S3 aktivieren, RTC während Sie [eine Regel erstellen oder bearbeiten](#). S3 RTC umfasst weitere Funktionen, wie z. B. ein Service Level Agreement (SLA) und Benachrichtigungen bei Überschreitung von Schwellenwerten. Weitere Informationen finden Sie unter [Erfüllung der Compliance-Anforderungen mit S3 Replication Time Control \(S3RTC\)](#).

Wenn S3-Replikationsmetriken aktiviert sind, veröffentlicht Amazon S3 die folgenden Metriken auf Amazon CloudWatch. CloudWatchDie Metriken werden nach bestem Wissen und Gewissen bereitgestellt.

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
Noch ausstehende Bytes zur Replikation	Die Gesamtanzahl der Bytes von Objekten, deren Replikation für eine bestimmte Replikationsregel aussteht.	Diese Metrik gilt nur für neue Objekte, die mit S3 Cross-Region Replication (S3CRR) oder S3 Same-Region Replication (S3) repliziert werden. SRR	Diese Metrik wird in der Region des Ziel-Buckets veröffentlicht.	Nein	Ja
Latenz der Replikation	Die maximale Anzahl von Sekunden, um die sich der Replizierungsziel-Bucket für eine bestimmte Replikationsregel hinter dem	Diese Metrik gilt nur für neue Objekte, die mit S3 CRR oder S3 SRR repliziert werden.	Diese Metrik wird in der Region des Ziel-Buckets veröffentlicht.	Nein	Ja

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
	Quell-Bucket befindet.				
Operationen, die noch repliziert	Die Anzahl der Operationen, deren Replikation für eine bestimmte Replikationsregel aussteht. Diese Metrik verfolgt Vorgänge im Zusammenhang mit Objekten, Löschmarkierungen, Tags, Zugriffskontrolllisten (ACLs) und S3 Object Lock.	Diese Metrik gilt nur für neue Objekte, die mit S3 CRR oder S3 SRR repliziert werden.	Diese Metrik wird in der Region des Ziel-Buckets veröffentlicht.	Nein	Ja

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
Operationen sind fehlgeschlagen, Replikation	Die Anzahl von Vorgängen, bei denen die Replikation für eine bestimmte Replikationsregel fehlgeschlagen ist. Diese Metrik verfolgt Vorgänge im Zusammenhang mit Objekten, Löschmarkierungen, Tags, Zugriffskontrolllisten (ACLs) und Objektsperren. Operationen sind fehlgeschlagen	Diese Metrik gilt sowohl für neue Objekte, die mit S3 CRR oder S3 SRR repliziert werden, als auch für vorhandene Objekte, die mit S3 Batch Replikation repliziert werden. <div data-bbox="592 1304 808 1869" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>Wenn ein S3-Batch-Replikationsauftrag überhaupt nicht ausgeführt</p> </div>	Diese Metrik wird in der Region des Quell-Buckets veröffentlicht.	Ja	Nein

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
	<p>lagen Replikation verfolgt S3-Replikationsfehler zusammengefasst in einem minutlichen Intervall. Um die spezifischen Objekte zu ermitteln, bei denen die Replikation fehlgeschlagen ist, und die Fehlerursachen herauszufinden, abonnieren Sie das Operation FailedReplication - Ereignis in den Amazon-</p>	<p>t werden kann, werden keine Metriken an Amazon gesendet CloudWatch. Ihr Auftrag wird beispielsweise nicht ausgeführt, wenn Sie nicht über die erforderlichen Berechtigungen verfügen,</p>			

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
	<p>S3-Ereignisbenachrichtigungen. Weitere Informationen finden Sie unter Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlereignisse.</p>	<p>um einen S3-Batch-Replikationsauftrag auszuführen, oder wenn die Tags oder das Präfix in Ihrer Replikationskonfiguration nicht übereinstimmen.</p>			

Informationen zur Arbeit mit diesen Metriken CloudWatch finden Sie unter [the section called “Metriken zur S3-Replikation in CloudWatch”](#).

Aktivieren von S3-Replikationsmetriken

Sie können S3-Replikationsmetriken mit einer neuen oder einer vorhandenen Replikationsregel verwenden. Alle Informationen zum Erstellen von Replikationsregeln finden Sie unter [Konfigurieren der Replikation für Quell- und Ziel-Buckets im Eigentum desselben Kontos](#). Sie können die Replikationsregel auf einen gesamten S3-Bucket oder auf Amazon-S3-Objekte mit einem bestimmten Präfix oder Tag anwenden.

Dieses Thema enthält Anweisungen zur Aktivierung von S3-Replikationsmetriken in Ihrer Replikationskonfiguration, wenn die Quell- und Ziel-Buckets demselben oder verschiedenen gehören AWS-Konten.

Um Replikationsmetriken zu aktivieren, verwenden Sie AWS Command Line Interface (AWS CLI), müssen Sie dem Quell-Bucket eine Replikationskonfiguration mit `Metrics` aktivierter Option hinzufügen. In dieser Beispielkonfiguration werden Objekte unter dem Präfix `Tax` auf den Ziel-Bucket `amzn-s3-demo-bucket` repliziert und es werden Metriken für diese Objekte generiert.

```
{
  "Rules": [
    {
      "Status": "Enabled",
      "Filter": {
        "Prefix": "Tax"
      },
      "Destination": {
        "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
        "Metrics": {
          "Status": "Enabled"
        }
      },
      "Priority": 1
    }
  ],
  "Role": "IAM-Role-ARN"
}
```

Anzeigen von Replikationsmetriken

Sie können S3-Replikationsmetriken auf der Registerkarte Metriken des Quell-Buckets in der Amazon S3 S3-Konsole anzeigen. Diese CloudWatch Amazon-Metriken sind auch in der CloudWatch Amazon-Konsole verfügbar. Wenn Sie S3-Replikationsmetriken aktivieren, CloudWatch gibt Amazon Metriken aus, mit denen Sie ausstehende Bytes, ausstehende Operationen und Replikationslatenz auf der Ebene der Replikationsregeln verfolgen können.

S3-Replikationsmetriken werden automatisch aktiviert, wenn Sie die Replikation mit S3 Replication Time Control (S3RTC) über die Amazon S3-Konsole oder Amazon S3 aktivieren RESTAPI. Sie können S3-Replikationsmetriken auch unabhängig von S3 aktivieren, RTC während Sie [eine Regel erstellen oder bearbeiten](#).

Wenn Sie S3 Replication Time Control verwenden, CloudWatch beginnt Amazon 15 Minuten, nachdem Sie S3 RTC für die jeweilige Replikationsregel aktiviert haben, mit der Meldung von Replikationsmetriken. Weitere Informationen finden Sie unter [Replikationsmetriken mit S3 RTC](#).

Replikationsmetriken verfolgen die Regel IDs der Replikationskonfiguration. Eine Replikationsregel-ID kann spezifisch für ein Präfix, ein Tag oder eine Kombination aus beiden sein.

Weitere Informationen zu CloudWatch Metriken für Amazon S3 finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

Voraussetzungen

Erstellen Sie eine Replikationsregel, für die S3-Replikationsmetriken aktiviert sind. Weitere Informationen finden Sie unter [the section called “Aktivieren von Replikationsmetriken”](#).

Um S3-Replikationsmetriken über den Tab Metriken des Quell-Buckets anzuzeigen

1. Melden Sie sich bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Allgemeine Buckets den Namen des Quell-Buckets aus, der die Objekte enthält, für die Sie Replikationsmetriken benötigen.
4. Wählen Sie die Registerkarte Metriken.
5. Wählen Sie unter Replikationsmetriken die Replikationsregeln aus, für die Sie Metriken anzeigen möchten.

6. Wählen Sie Display charts (Diagramme anzeigen).

Amazon S3 zeigt für die von Ihnen ausgewählten Regeln Diagramme für die Replikationslatenz, die Anzahl der zur Replikation ausstehenden Byte, zur Replikation ausstehende Vorgänge und zur fehlgeschlagenen Replikation von Vorgängen an.

Replikationsmetriken in den Dashboards von S3 Storage Lens anzeigen

Zusätzlich zu den [S3-Replikationsmetriken](#) können Sie die replikationsbezogenen Datenschutzmetriken verwenden, die von S3 Storage Lens bereitgestellt werden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter [Verwenden von S3 Storage Lens zum Schutz Ihrer Daten](#).

S3 Storage Lens bietet zwei Stufen von Metriken: kostenlose Metriken und erweiterte Metriken und Empfehlungen, auf die Sie gegen Aufpreis ein Upgrade durchführen können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen, um Erkenntnisse über Ihren Speicher zu erhalten. Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter [Amazon S3-Preise](#).

Wenn Sie die kostenlosen Metriken in S3 Storage Lens verwenden, können Sie sich Metriken wie die Gesamtzahl der Byte, die aus dem Quell-Bucket repliziert wurden, oder die Anzahl der replizierten Objekte aus dem Quell-Bucket anzeigen lassen.

Um Ihre allgemeine Einstellung zur Replikation zu überprüfen, können Sie erweiterte Metriken in S3 Storage Lens aktivieren. Mit den erweiterten Metriken in S3 Storage Lens können Sie sehen, über wie viele Replikationsregeln Sie verschiedener Typen verfügen, einschließlich der Anzahl der Replikationsregeln mit einem ungültigen Replikationsziel.

Eine vollständige Liste der S3 Storage Lens-Metriken, einschließlich der Replikationsmetriken in den einzelnen Stufen, finden Sie im [Glossar zu S3 Storage Lens-Metriken](#).

Voraussetzungen


Erstellen Sie eine [Live-Replikationskonfiguration](#) oder einen [S3-Batch-Replikationsauftrag](#).

So zeigen Sie Replikationsmetriken in Amazon S3 Storage Lens an

1. Erstellen Sie ein S3 Storage Lens-Dashboard. step-by-step Anweisungen finden Sie unter [the section called "Verwenden der S3-Konsole"](#).

2. (Optional) Wenn Sie bei der Einrichtung Ihres Dashboards alle Replikationsmetriken für S3 Storage Lens sehen möchten, wählen Sie Erweiterte Metriken und Empfehlungen und dann Erweiterte Datenschutzmetriken aus. Eine vollständige Liste der Metriken finden Sie im [Glossar zu S3 Storage Lens-Metriken](#).

Wenn Sie erweiterte Metriken und Empfehlungen aktivieren, können Sie weitere Einblicke in Ihre Replikationskonfigurationen gewinnen. Sie können beispielsweise Messwerte zur Anzahl der Replikationsregeln von S3 Storage Lens verwenden, um detaillierte Informationen zu Ihren Buckets zu erhalten, die für die Replikation konfiguriert sind. Zu diesen Informationen gehören Replikationsregeln innerhalb von Buckets und Regionen sowie Bucket- und regionsübergreifende Replikationsregeln. Weitere Informationen finden Sie unter [the section called "Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket"](#).

3. Nachdem Sie Ihr Dashboard erstellt haben, öffnen Sie das Dashboard und wählen Sie den Tab Buckets.
4. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus. Löschen Sie dann Summary (Zusammenfassung).
5. Um die Buckets-Liste so zu filtern, dass nur Replikationsmetriken angezeigt werden, wählen Sie das Einstellungen-Symbol

6. Deaktivieren Sie die Schalter für alle Datenschutzmetriken, bis nur noch die Replikationsmetriken ausgewählt sind.
7. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
8. Klicken Sie auf Weiter.

Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse

Wenn Sie S3-Replikationsmetriken in Ihrer Replikationskonfiguration aktiviert haben, können Sie Amazon S3 S3-Ereignisbenachrichtigungen einrichten, um Sie zu benachrichtigen, wenn Objekte nicht an ihr Ziel repliziert werden AWS-Region. Wenn Sie S3 Replication Time Control (S3RTC) in Ihrer Replikationskonfiguration aktiviert haben, können Sie auch benachrichtigt werden, wenn Objekte nicht innerhalb des RTC S3-Schwellenwerts von 15 Minuten repliziert werden.

Mithilfe der folgenden Replication Ereignistypen können Sie den minute-by-minute Fortschritt von Replikationsereignissen überwachen, indem Sie ausstehende Bytes, ausstehende Vorgänge und die Replikationslatenz verfolgen. Weitere Informationen zu S3-Replikationsmetriken finden Sie unter [Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status](#).

- Der `s3:Replication:OperationFailedReplication`-Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation berechtigt war, nicht repliziert werden konnte.
- Der `s3:Replication:OperationMissedThreshold` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, den 15-minütigen Schwellenwert für die Replikation RTC überschreitet.
- Der `s3:Replication:OperationReplicatedAfterThreshold` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, nach dem RTC 15-Minuten-Schwellenwert repliziert wird.
- Der `s3:Replication:OperationNotTracked` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Live-Replikation in Frage kam (entweder regionsübergreifende Replikation [SRR] oder regionsübergreifende Replikation [CRR]), nicht mehr anhand von Replikationsmetriken verfolgt wird.

Vollständige Beschreibungen aller unterstützten Replikationsereignistypen finden Sie unter [the section called “Unterstützte Ereignistypen für SQSSNS, und Lambda”](#)

Eine Liste der von S3-Ereignisbenachrichtigungen erfassten Fehlercodes finden Sie unter [Gründe für das Fehlschlagen der Replikation in Amazon S3](#).

Sie können S3-Ereignisbenachrichtigungen über Amazon Simple Queue Service (AmazonSQS), Amazon Simple Notification Service (AmazonSNS) erhalten, oder AWS Lambda. Weitere Informationen finden Sie unter [the section called “Amazon-S3-Ereignis-Benachrichtigungen”](#).

Anweisungen zum Konfigurieren von Amazon-S3-Ereignisbenachrichtigungen finden Sie unter [Aktivieren von Ereignisbenachrichtigungen](#).

Note

Stellen Sie sicher, dass Sie nicht nur Ereignisbenachrichtigungen aktivieren, sondern auch S3-Replikationsmetriken aktivieren. Weitere Informationen finden Sie unter [the section called “Aktivieren von Replikationsmetriken”](#).

Das Folgende ist ein Beispiel für eine Nachricht, die Amazon S3 sendet, um ein `s3:Replication:OperationFailedReplication`-Ereignis zu veröffentlichen. Weitere Informationen finden Sie unter [the section called "Struktur von Ereignismeldungen"](#).

```
{
  "Records": [
    {
      "eventVersion": "2.2",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "2024-09-05T21:04:32.527Z",
      "eventName": "Replication:OperationFailedReplication",
      "userIdentity": {
        "principalId": "s3.amazonaws.com"
      },
      "requestParameters": {
        "sourceIPAddress": "s3.amazonaws.com"
      },
      "responseElements": {
        "x-amz-request-id": "123bf045-2b4b-4ca8-a211-c34a63c59426",
        "x-amz-id-2":
"12VAWNDIHNwJsRhTccqQTeAPoXQmRt22KkewMV8G3XZihAuf9CLDdmkApgZzudaIe2K1LfdqGS0="
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "ReplicationEventName",
        "bucket": {
          "name": "amzn-s3-demo-bucket1",
          "ownerIdentity": {
            "principalId": "111122223333"
          },
          "arn": "arn:aws:s3:::amzn-s3-demo-bucket1"
        },
        "object": {
          "key": "replication-object-put-test.png",
          "size": 520080,
          "eTag": "e12345ca7e88a38428305d3ff7fcb99f",
          "versionId": "abcdeH0Xp66ep__QDjR76LK7Gc9X4wK0",
          "sequencer": "0066DA1CBF104C0D51"
        }
      },
      "replicationEventData": {
        "replicationRuleId": "notification-test-replication-rule",

```



```

    "destinationBucket": "arn:aws:s3:::amzn-s3-demo-bucket2",
    "s3Operation": "OBJECT_PUT",
    "requestTime": "2024-09-05T21:03:59.168Z",
    "failureReason": "AssumeRoleNotPermitted"
  }
}
]
}
```

Gründe für das Fehlschlagen der Replikation in Amazon S3

Die folgende Tabelle enthält Gründe für das Fehlschlagen der Replikation in Amazon S3. Sie können sich diese Gründe ansehen, indem Sie das `s3:Replication:OperationFailedReplication` Ereignis mit Amazon S3 S3-Ereignisbenachrichtigungen erhalten und sich dann den `failureReason` Wert ansehen.

Sie können diese Fehlergründe auch in einem Abschlussbericht der S3-Batch-Replikation anzeigen. Weitere Informationen finden Sie unter [Abschlussbericht zur Batchreplikation](#).

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>AssumeRoleNotPermitted</code>	Amazon S3 kann das nicht annehmen AWS Identity and Access Management (IAM) Rolle, die in der Replikationskonfiguration oder im Batch Operations-Job angegeben ist.
<code>DstBucketInvalidRegion</code>	Der Ziel-Bucket befindet sich nicht im selben AWS-Region wie im Batch Operations-Job angegeben. Dieser Fehler ist spezifisch für die Batchreplikation.
<code>DstBucketNotFound</code>	Amazon S3 kann den in der Replikationskonfiguration angegebenen Ziel-Bucket nicht finden.
<code>DstBucketObjectLockConfigMissing</code>	Um Objekte aus einem Quell-Bucket mit aktivierter Objektsperre zu replizieren, muss für den Ziel-Bucket ebenfalls die Objektsperre aktiviert sein. Dieser Fehler zeigt an, dass die Objektsperre im Ziel-Bucket möglicher

Gründe für das Fehlschlagen der Replikation	Beschreibung
	weise nicht aktiviert ist. Weitere Informationen finden Sie unter Überlegungen zu Object Lock .
<code>DstBucketUnversioned</code>	Die Versionsverwaltung ist für den S3-Ziel-Bucket nicht aktiviert. Aktivieren Sie die Versionsverwaltung für den Ziel-Bucket, um Objekte mit S3-Replikation zu replizieren.
<code>DstDelObjNotPermitted</code>	Amazon S3 kann Löschmarkierungen nicht in den angegebenen Ziel-Bucket replizieren. Möglicherweise fehlt die <code>s3:ReplicateDelete</code> -Berechtigung für den Ziel-Bucket.
<code>DstKmsKeyInvalidState</code>	Das Tool AWS Key Management Service (AWS KMS) Der Schlüssel für den Ziel-Bucket hat keinen gültigen Status. Überprüfen und aktivieren Sie die erforderlichen AWS KMS Schlüssel. Für weitere Informationen zur Verwaltung AWS KMS Schlüssel finden Sie unter Wichtigste Zustände von AWS KMS Schlüssel in der AWS Key Management Service Leitfaden für Entwickler.
<code>DstKmsKeyNotFound</code>	Das Tool AWS KMS Der Schlüssel, der für den Ziel-Bucket in der Replikationskonfiguration konfiguriert ist, ist nicht vorhanden.
<code>DstMultipartCompleteNotPermitted</code>	Amazon S3 kann mehrteilige Uploads von Objekten im Ziel-Bucket nicht abschließen. Möglicherweise fehlt die <code>s3:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>DstMultipartInitNotPermitted</code>	Amazon S3 kann mehrteilige Uploads von Objekten in den Ziel-Bucket nicht initiieren. Möglicherweise fehlt die <code>s3:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstMultipartPartUploadNotPermitted</code>	Amazon S3 kann keine mehrteiligen Objekte in den Ziel-Bucket hochladen. Möglicherweise fehlt die <code>s3:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstObjectHardDeleted</code>	Die S3-Batchreplikation unterstützt nicht das erneute Replizieren von Objekten, die mit der Versions-ID des Objekts aus dem Ziel-Bucket gelöscht wurden. Dieser Fehler ist spezifisch für die Batchreplikation.
<code>DstPutAclNotPermitted</code>	Amazon S3 ist nicht in der Lage, Objektzugriffskontrolllisten (ACLs) in den Ziel-Bucket zu replizieren. Möglicherweise fehlt die <code>s3:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstPutLegalHoldNotPermitted</code>	Amazon S3 kann keine Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen für die Zielobjekte festlegen, während unveränderliche Objekte repliziert werden. Möglicherweise fehlt die <code>s3:PutObjectLegalHold</code> -Berechtigung für den Ziel-Bucket. Weitere Informationen finden Sie unter Rechtliche Aufbewahrungsfristen .

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>DstPutObjectNotPermitted</code>	Amazon S3 kann Objekte nicht in den angegebenen Ziel-Bucket replizieren. Möglicherweise fehlen die <code>s3:ReplicateObject</code> - oder <code>s3:ObjectOwnerOverrideToBucketOwner</code> - Berechtigungen für den Ziel-Bucket.
<code>DstPutTaggingNotPermitted</code>	Amazon S3 kann Objekt-Tags nicht in den angegebenen Ziel-Bucket replizieren. Möglicherweise fehlt die <code>s3:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstVersionNotFound</code>	Amazon S3 kann die erforderliche Objektversion im Ziel-Bucket nicht finden, für den Metadaten repliziert werden müssen.
<code>InitiateReplicationNotPermitted</code>	Amazon S3 kann keine Replikation für Objekte initiieren. Möglicherweise fehlt die <code>s3:InitiateReplication</code> -Berechtigung für den Batch-Operationsauftrag. Dieser Fehler ist spezifisch für die Batchreplikation.
<code>SrcBucketInvalidRegion</code>	Der Quell-Bucket befindet sich nicht in der selben AWS-Region wie im Batch Operations-Job angegeben. Dieser Fehler ist spezifisch für die Batchreplikation.
<code>SrcBucketNotFound</code>	Amazon S3 kann den Quell-Bucket nicht finden.
<code>SrcBucketReplicationConfigMissing</code>	Amazon S3 konnte keine Replikationskonfiguration für den Quell-Bucket finden.

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>SrcGetAclNotPermitted</code>	<p>Amazon S3 kann nicht auf das Objekt im Quell-Bucket für die Replikation zugreifen . Möglicherweise fehlt die <code>s3:GetObjectVersionAcl</code> -Berechtigung für das Quell-Bucket-Objekt.</p> <p>Die Objekte im Quell-Bucket müssen sich im Besitz des Bucket-Eigentümers befinden. Falls ACLs aktiviert, überprüfen Sie, ob „Objektbesitz“ auf „Bucket Owner Preferred“ oder „Object Writer“ gesetzt ist. Wenn Object Ownership auf Bucket Owner Preferred gesetzt ist, müssen die Quell-Bucket-Objekte die Eigenschaft <code>bucket-owner-full-control</code> ACL haben, damit der Bucket-Besitzer zum Objekteigentümer wird. Das Quellkonto kann das Eigentum an allen Objekten in seinem Bucket übernehmen, indem es die Option Objekteigentum auf Bucket Ownerforced setzt und deaktiviert ACLs.</p>
<code>SrcGetLegalHoldNotPermitted</code>	<p>Amazon S3 kann nicht auf die Informationen zur gesetzlichen Aufbewahrung von S3 Object Lock zugreifen.</p>
<code>SrcGetObjectNotPermitted</code>	<p>Amazon S3 kann nicht auf das Objekt im Quell-Bucket für die Replikation zugreifen . Möglicherweise fehlt die <code>s3:GetObjectVersionForReplication</code> -Berechtigung für den Quell-Bucket.</p>
<code>SrcGetRetentionNotPermitted</code>	<p>Amazon S3 kann nicht auf die Informationen zum Aufbewahrungszeitraum von S3 Object Lock zugreifen.</p>

Gründe für das Fehlschlagen der Replikation	Beschreibung
SrcGetTaggingNotPermitted	Amazon S3 kann nicht auf Objekt-Tag-Informationen aus dem Quell-Bucket zugreifen. Möglicherweise fehlt die <code>s3:GetObjectVersionTagging</code> -Berechtigung für den Quell-Bucket.
SrcHeadObjectNotPermitted	Amazon S3 kann keine Objektmetadaten aus dem Quell-Bucket abrufen. Möglicherweise fehlt die <code>s3:GetObjectVersionForReplication</code> -Berechtigung für den Quell-Bucket.
SrcKeyNotFound	Amazon S3 kann den Quellobjektschlüssel für die Replikation nicht finden. Das Quellobjekt wurde möglicherweise gelöscht, bevor die Replikation abgeschlossen war.
SrcKmsKeyInvalidState	Das Tool AWS KMS Der Schlüssel für den Quell-Bucket hat keinen gültigen Status. Überprüfen und aktivieren Sie die erforderlichen AWS KMS Schlüssel. Für weitere Informationen zur Verwaltung AWS KMS Schlüssel finden Sie unter Wichtigste Zustände von AWS KMS Schlüssel in der AWS Key Management Service Leitfaden für Entwickler.
SrcObjectNotEligible	Einige Objekte kommen nicht für die Replikation infrage. Dies kann an der Speicherklasse des Objekts liegen oder daran, dass die Objekt-Tags nicht mit der Replikationskonfiguration übereinstimmen.
SrcObjectNotFound	Das Quellobjekt ist nicht vorhanden.

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>SrcReplicationNotPending</code>	Amazon S3 hat dieses Objekt bereits repliziert. Für dieses Objekt steht keine Replikation mehr aus.
<code>SrcVersionNotFound</code>	Amazon S3 kann den Quellobjektversion für die Replikation nicht finden. Die Quellobjektversion wurde möglicherweise gelöscht, bevor die Replikation abgeschlossen war.

Verwandte Themen

[Berechtigungen für die Live-Replikation einrichten](#)

[Fehlerbehebung bei einer Replikation](#)

Abrufen von Replikationsstatusinformationen

Der Replikationsstatus hilft Ihnen, den aktuellen Status eines derzeit replizierten Objekts zu bestimmen. Der Replikationsstatus eines Quellobjekts gibt entweder PENDING, COMPLETED oder FAILED zurück. Der Replikationsstatus eines Replikats gibt REPLICA zurück.

Sie können auch Werte für den Replikationsstatus verwenden, wenn Sie S3-Batch-Replikationsaufträge erstellen. Sie können diese Statuswerte beispielsweise verwenden, um Objekte zu replizieren, die entweder nie repliziert wurden oder deren Replikation fehlgeschlagen ist. Weitere Hinweise zur Verwendung dieser Werte mit der Batch-Replikation finden Sie unter [the section called "Verwenden von Replikationsstatusinformationen mit Batch-Replikationsaufträgen"](#).

Themen

- [Übersicht über den Replikationsstatus](#)
- [Replikationsstatus bei einer Replikation auf mehrere Ziel-Buckets](#)
- [Replikationsstatus, wenn die Synchronisierung von Amazon-S3-Replikatänderungen aktiviert ist](#)
- [Verwenden von Replikationsstatusinformationen mit Batch-Replikationsaufträgen](#)
- [Finden des Replikationsstatus](#)

Übersicht über den Replikationsstatus

Bei der Replikation haben Sie einen Quell-Bucket, in dem Sie die Replikation konfigurieren, und einen oder mehrere Ziel-Buckets, in denen Amazon S3 Objekte repliziert. Wenn Sie ein Objekt (mithilfe `GetObject`) oder Objektmetadaten (mithilfe von `HeadObject`) aus diesen Buckets anfordern, gibt Amazon S3 den `x-amz-replication-status` Header in der Antwort zurück:


- Wenn Sie ein Objekt aus dem Quell-Bucket anfordern, gibt Amazon S3 den Header `x-amz-replication-status` zurück, wenn das Objekt in der Anforderung für die Replikation geeignet ist.

Nehmen wir beispielsweise an, dass Sie in Ihrer Replikationskonfiguration das Objektpräfix `TaxDocs` angeben, um Amazon S3 anzuweisen, nur Objekte mit dem Schlüsselnamenpräfix `TaxDocs` zu replizieren. Alle Objekte mit diesem Schlüsselnamenpräfix, die Sie hochladen, z. B. `TaxDocs/document1.pdf`, werden repliziert. Für Objektanforderungen mit diesem Schlüsselnamenpräfix gibt Amazon S3 den Header `x-amz-replication-status` mit einem der folgenden Werte für den Replikationsstatus des Objekts zurück: `PENDING`, `COMPLETED` oder `FAILED`.

Note

Wenn nach dem Hochladen eines Objekts die Objektreplikation fehlschlägt, können Sie die fehlgeschlagene Replikation nicht erneut durchzuführen versuchen. Sie müssen das Objekt erneut hochladen, oder Sie müssen S3 Batch Replication verwenden, um alle fehlgeschlagenen Objekte zu replizieren. Weitere Informationen zur Verwendung der Batch-Replikation finden Sie unter [the section called “Replizieren vorhandener Objekte”](#). Objekte wechseln aufgrund von Problemen wie fehlenden Rollenberechtigungen für die Replikation in einen bestimmten `FAILED` Status, AWS Key Management Service (AWS KMS) Berechtigungen oder Bucket-Berechtigungen. Bei vorübergehenden Ausfällen, z. B. wenn ein Bucket oder eine Region nicht verfügbar ist, wechselt der Replikationsstatus nicht zu `FAILED`, sondern bleibt bestehen `PENDING`. Nachdem die Ressource wieder online ist, setzt Amazon S3 die Replikation dieser Objekte fort.

- Wenn Sie ein Objekt aus einem Ziel-Bucket anfordern und es sich bei dem Objekt Ihrer Anforderung um ein Replikat handelt, das Amazon S3 erstellt hat, gibt Amazon S3 den Header `x-amz-replication-status` mit dem Wert `REPLICA` zurück.

 Note

Bevor Sie ein Objekt aus einem Quell-Bucket löschen, für das die Replikation aktiviert ist, überprüfen Sie den Replikationsstatus des Objekts, um sicherzustellen, dass das Objekt repliziert wurde.

Wenn eine S3-Lifecycle-Konfiguration im Quell-Bucket aktiviert ist, setzt Amazon S3 Lebenszyklusaktionen aus, bis der Status der Objekte entweder als oder COMPLETED markiert wird. FAILED

Replikationsstatus bei einer Replikation auf mehrere Ziel-Buckets

Wenn Sie Objekte in mehrere Ziel-Buckets replizieren, verhält sich der Header `x-amz-replication-status` anders. Der Header des Quellobjekts gibt COMPLETED nur dann den Wert zurück, wenn die Replikation an alle Ziele erfolgreich ist. Der Header bleibt auf dem Wert PENDING, bis die Replikation für alle Ziele abgeschlossen ist. Wenn die Replikation bei einem oder mehreren Zielen fehlschlägt, liefert der Header FAILED.

Replikationsstatus, wenn die Synchronisierung von Amazon-S3-Replikatänderungen aktiviert ist

Wenn in Ihren Replikationsregeln die Synchronisierung von Amazon-S3-Replikatänderungen aktiviert ist, können Replikate einen anderen Status als REPLICIA melden. Wenn Änderungen an Metadaten gerade repliziert werden, gibt der `x-amz-replication-status`-Header den Wert PENDING zurück. Wenn die Synchronisierung der Replikatänderung Metadaten nicht repliziert, gibt der Header FAILED zurück. Wenn Metadaten korrekt repliziert werden, geben die Replikate den Header REPLICIA zurück.

Verwenden von Replikationsstatusinformationen mit Batch-Replikationsaufträgen

Wenn Sie einen Batch-Replikationsauftrag erstellen, können Sie optional zusätzliche Filter angeben, z. B. das Erstellungsdatum des Objekts und den Replikationsstatus, um den Umfang des Jobs zu reduzieren.

Sie können Objekte filtern, die repliziert werden sollen, basierend auf dem `ObjectReplicationStatuses`-Wert, indem Sie einen oder mehrere der folgenden Werte angeben:

- "NONE" – Zeigt an, dass Amazon S3 noch nie versucht hat, das Objekt zu replizieren.

- "FAILED"— Zeigt an, dass Amazon S3 versucht hat, das Objekt zuvor zu replizieren, dies jedoch fehlgeschlagen ist.
- "COMPLETED" – Zeigt an, dass Amazon S3 das Objekt zuvor erfolgreich repliziert hat.
- "REPLICA"— Zeigt an, dass es sich um ein Replikat-Objekt handelt, das Amazon S3 aus einer anderen Quelle repliziert hat.

Weitere Hinweise zur Verwendung dieser Replikationsstatuswerte mit der Batch-Replikation finden Sie unter [the section called “Filter für einen Batch-Replikationsauftrag”](#).

Finden des Replikationsstatus

Verwenden Sie das Amazon-S3-Inventory-Tool, um den Replikationsstatus der Objekte in einem Bucket abzurufen. Amazon S3 sendet eine CSV Datei an den Ziel-Bucket, den Sie in der Inventarkonfiguration angeben. Sie können auch Amazon Athena verwenden, um den Replikationsstatus im Bestandsbericht abzufragen. Weitere Informationen zu Amazon S3 Inventory finden Sie unter [Amazon S3 Inventory](#).

Sie können den Status der Objektreplikation auch mithilfe der Amazon S3 S3-Konsole ermitteln, AWS Command Line Interface (AWS CLI) oder AWS SDK.

Verwenden der S3-Konsole

In der Amazon S3 S3-Konsole können Sie den Replikationsstatus für ein Objekt auf der Detailseite des Objekts einsehen.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste der Buckets für allgemeine Zwecke den Namen des Replizierungsquellen-Buckets aus.
4. Wählen Sie in der Liste Objekte den Objektnamen aus. Die Seite mit den Objektdetails wird angezeigt.
5. Scrollen Sie auf der Registerkarte Eigenschaften nach unten zum Abschnitt Übersicht über die Objektverwaltung. Unter Verwaltungskonfigurationen finden Sie den Wert unter Replikationsstatus.

Verwendung der AWS CLI

Verwenden Sie den AWS Command Line Interface (AWS CLI) `head-object`-Befehl, um Objektmetadaten abzurufen, wie im folgenden Beispiel gezeigt. Ersetzen Sie den *amzn-s3-demo-source-bucket1* durch den Namen Ihres Replizierungsquellen-Buckets und den anderen *user input placeholders* durch Ihre eigenen Informationen.

```
aws s3api head-object --bucket amzn-s3-demo-source-bucket1 --key object-key --version-id object-version-id
```

Der Befehl gibt Informationen zu Objektmetadaten, einschließlich des `ReplicationStatus`, zurück, wie in der folgenden Beispielantwort dargestellt:

```
{
  "AcceptRanges": "bytes",
  "ContentType": "image/jpeg",
  "LastModified": "Mon, 23 Mar 2015 21:02:29 GMT",
  "ContentLength": 3191,
  "ReplicationStatus": "COMPLETED",
  "VersionId": "jfnW.HIM0fYiD_9rGbSkmroXsFj3fqZ.",
  "ETag": "\"6805f2cfc46c0f04559748bb039d69ae\"",
  "Metadata": {}
}
```

Verwendung der AWS SDKs

Die folgenden Codefragmente ermitteln Ihren Replikationsstatus mithilfe der AWS SDK for Java and AWS SDK for .NET jeweils.

Java

```
GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest(bucketName,
    key);
ObjectMetadata metadata = s3Client.getObjectMetadata(metadataRequest);

System.out.println("Replication Status : " +
    metadata.getRawMetadataValue(Headers.OBJECT_REPLICATION_STATUS));
```

.NET

```
GetObjectMetadataRequest getmetadataRequest = new GetObjectMetadataRequest
{
    BucketName = sourceBucket,
    Key        = objectKey
};

GetObjectMetadataResponse getmetadataResponse =
    client.GetObjectMetadata(getmetadataRequest);
Console.WriteLine("Object replication status: {0}",
    getmetadataResponse.ReplicationStatus);
```

Kategorisieren des Speichers mithilfe von Markierungen

Markieren Sie Objekte, um Speicher zu kategorisieren. Jeder Tag ist ein Schlüssel/Wert-Paar.

Sie können neuen Objekten Markierungen hinzufügen, wenn Sie sie hochladen, Sie können Markierungen aber auch vorhandenen Objekten hinzufügen.

- Sie können einem Objekt bis zu 10 Markierungen zuordnen. Einem Objekt zugeordnete Markierungen müssen eindeutige Tag-Schlüssel haben.
- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Amazon-S3-Objekt-Tags werden intern in UTF-16 dargestellt. Beachten Sie, dass Zeichen in UTF-16 entweder 1 oder 2 Zeichenpositionen einnehmen.
- Bei Schlüssel und Werten wird die Groß-/Kleinschreibung berücksichtigt.
- Weitere Informationen zu Tagbeschränkungen finden Sie unter [Benutzerdefinierte Tagbeschränkungen](#) im AWS Billing and Cost Management-Benutzerhandbuch. Grundlegende Tag-Einschränkungen finden Sie unter [Tag-Einschränkungen](#) im Amazon EC2 EC2-Benutzerhandbuch.

Beispiele

Betrachten Sie die folgenden Beispiele für die Markierung:

Example PHI-Informationen

Angenommen, ein Objekt enthält PHI-Daten (Protected Health Information, geschützte Gesundheitsdaten). Sie könnten das Objekt unter Verwendung des folgenden-Schlüssel-Wert-Paares markieren:

```
PHI=True
```

oder

```
Classification=PHI
```

Example Projektdateien

Angenommen, Sie speichern Projektdateien in Ihrem S3-Bucket. Sie könnten diese Objekte mit einem Schlüssel namens `Project` und einem Wert markieren, wie nachfolgend gezeigt:

```
Project=Blue
```

Example Mehrere Markierungen

Sie können einem Objekt mehrere Markierungen hinzufügen, wie nachfolgend gezeigt:

```
Project=x  
Classification=confidential
```

Schlüsselnamen-Präfixe und -Markierungen

Mit Schlüsselnamenpräfixe können Sie auch Speicher kategorisieren. Allerdings sind Präfix-basierte Kategorisierungen eindimensional. Sehen Sie sich die folgenden Objektschlüsselnamen an:

```
photos/photo1.jpg  
project/projectx/document.pdf  
project/projecty/document2.pdf
```

Dieses Schlüsselnamen haben die Präfixe `photos/`, `project/projectx/` und `project/projecty/`. Diese Präfixe unterstützen eine eindimensionale Kategorisierung. Das bedeutet, alles unter einem Präfix ist eine Kategorie. Beispielsweise identifiziert das Präfix `project/projectx` alle Dokumente, die zu Projekt X gehören.

Mit der Markierung erhalten Sie jetzt eine weitere Dimension. Wenn Sie photo1 in der Kategorie project x anlegen wollen, können Sie das Objekt entsprechend markieren.

Zusätzliche Vorteile

Neben der Datenklassifizierung bietet die Markierung auch noch weitere Vorteile.

- Objekt-Markierungen bieten eine differenzierte Zugriffskontrolle für Berechtigungen. Sie könnten z. B. einem Benutzer Berechtigungen erteilen, nur Objekte mit bestimmten Markierungen zu lesen.
- Objekt-Markierungen unterstützen ein differenziertes Objektlebenszyklusmanagement, bei dem Sie in einer Lebenszyklusregel zusätzlich zum Schlüsselnamenpräfix einen auf Markierungen basierenden Filter angeben können.
- Wenn Sie Amazon-S3-Analysen verwenden, können Sie Filter konfigurieren, um Objekte für die Analyse nach Objekt-Markierungen, nach Schlüsselnamen-Präfix oder nach Präfix und Markierungen zu gruppieren.
- Sie können CloudWatch Amazon-Metriken auch so anpassen, dass Informationen anhand bestimmter Tag-Filter angezeigt werden. Die folgenden Abschnitte stellen Details bereit.

Important

Es ist akzeptabel, Markierungen zu verwenden, um Objekte mit vertraulichen Daten zu markieren (z. B. personenbezogene Informationen (PII) oder geschützte Gesundheitsinformationen (PHI)). Diese Markierungen sollten jedoch selbst keine vertraulichen Daten enthalten.

Hinzufügen von Objekt-Tag-Sätzen zu mehreren Amazon-S3-Objekten mit einer einzigen Anfrage

Zum Hinzufügen von Objekt-Tag-Mengen zu mehr als einem Amazon-S3-Objekt mit einer einzelnen Anforderung können Sie S3-Batchoperationen verwenden. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, für die Vorgänge ausgeführt werden sollen. S3-Batchoperationen rufen die entsprechende API-Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Die Funktion „S3-Batchoperationen“ verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. Sie profitieren von einer vollständig verwalteten, prüfbar und serverlosen Umgebung. Sie können S3 Batch Operations über die

Amazon S3 S3-Konsole AWS CLI, AWS SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [the section called “Grundlagen von BatchVorgänge”](#).

Weitere Informationen über Objekt-Markierungen finden Sie unter [Verwalten von Objekt-Markierungen](#).

API-Operationen für die Objektmarkierung

Amazon S3 unterstützt die folgenden API-Operationen, die spezifisch für das Objekt-Tagging sind:

Objekt-API-Operationen

- [PUT Object tagging](#) – Ersetzt Markierungen auf einem Objekt. Sie geben die Markierungen im Anfragerumpf an. Es gibt zwei unterschiedliche Szenarien der Objekt-Tag-Verwaltung unter Verwendung dieser API.
 - Objekt hat keine Markierungen – Mit Hilfe dieser API können Sie einem Objekt verschiedene Markierungen hinzufügen (das Objekt hat keine vorherigen Markierungen).
 - Das Objekt hat eine Menge vorhandener Markierungen – Um die vorhandene Tag-Menge zu ändern, müssen Sie zuerst die vorhandene Tag-Menge abrufen, sie auf der Client-Seite ändern und diese API dann verwenden, um die Tag-Menge zu ersetzen.

Note

Wenn Sie diese Anforderung mit einer leeren Tag-Menge senden, löscht Amazon S3 die vorhandene Tag-Menge für das Objekt. Wenn Sie diese Methode verwenden, wird eine Tier 1-Anforderung (PUT) berechnet. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Die Anforderung [DELETE Object tagging](#) wird empfohlen, weil sie das gleiche Ergebnis liefert, aber keine Kosten verursacht.

- [GET Object tagging](#) – Gibt die einem Objekt zugeordnete Tag-Menge zurück. Amazon S3 gibt die Objekt-Markierungen im Antworttext zurück.
- [DELETE Object tagging](#) – Löscht die mit einem Tag verbundenen Tag-Menge.

Andere API-Operationen, die Markieren unterstützen

- [PUT Object](#) und [Initiate Multipart Upload](#)– Sie können beim Erstellen von Objekten Markierungen angeben. Sie geben Markierungen unter Verwendung des Anfrage-Headers `x-amz-tagging` an.

- [GET Object](#) – Statt die Tag-Menge zurückzugeben, gibt Amazon S3 den Objekt-Tag-Zähler im `x-amz-tag-count`-Header zurück (nur dann, wenn der Auftraggeber die Berechtigung hat, Markierungen zu lesen), weil die Header-Antwortgröße auf 8 KB begrenzt ist. Wenn Sie die Markierungen anzeigen möchten, erstellen Sie eine weitere Anfrage für die API-Operation [GET Object tagging](#).
- [POST Object](#) – Sie können Markierungen in Ihrer POST-Anfrage angeben.

So lange die von Ihnen geforderten Markierungen die 8-KB-Größenbeschränkung der HTTP-Anfrageheader nicht überschreitet, können Sie die `PUT Object`-API verwenden, um Objekte mit Markierungen zu erstellen. Wenn die von Ihnen angegebenen Markierungen die Größenbeschränkung des Headers überschreiten, können Sie diese POST-Methode verwenden, wobei Sie die Markierungen in den Rumpf aufnehmen.

[PUT Object - Copy](#) – Sie können die `x-amz-tagging-directive` in Ihre Anfrage aufnehmen, um Amazon S3 anzuweisen, die Markierungen zu kopieren (Standardverhalten) oder durch eine neue, in der Anfrage angegebene Tag-Menge zu ersetzen.

Beachten Sie Folgendes:

- Die S3-Objektmarkierung ist hochgradig konsistent. Weitere Informationen finden Sie unter [Amazon S3-Datenkonsistenzmodell](#).

Zusätzliche Konfigurationen

Dieser Abschnitt erklärt, was die Objektmarkierung für andere Konfigurationen bedeutet.

Objektmarkierung und Lebenszyklusverwaltung

In der Bucket-Lebenszyklus-Konfiguration können Sie einen Filter angeben, um eine Untermenge von Objekten auszuwählen, auf die die Regel anzuwenden ist. Sie können einen Filter basierend auf den Schlüsselnamenpräfixen, Objekt-Markierungen oder beidem angeben.

Angenommen, Sie speichern Fotos (Rohdaten und im fertigen Format) in Ihrem Amazon-S3-Bucket. Sie könnten diese Objekte wie folgt markieren:

```
phototype=raw  
or  
phototype=finished
```


Sie könnten festlegen, dass die Rohdaten der Fotos irgendwann nach der Erstellung in S3 Glacier archiviert werden. Sie können eine Lebenszyklusregel mit einem Filter konfigurieren, der die Untermenge der Objekte mit dem Schlüsselnamenpräfix (photos/) identifiziert, die ein spezifisches Tag (phototype=raw) haben.

Weitere Informationen finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Objektmarkierung und -replikation

Wenn Sie auf Ihrem Bucket die Replikation konfiguriert haben, repliziert Amazon S3 Markierungen, sofern Sie Amazon S3 die Berechtigung zum Lesen der Markierungen erteilt haben. Weitere Informationen finden Sie unter [Live-Replikation einrichten](#).

Ereignisbenachrichtigungen einer Objektmarkierung

Sie können eine Amazon-S3-Ereignisbenachrichtigung einrichten, um benachrichtigt zu werden, wenn eine Objektmarkierung hinzugefügt oder aus einem Objekt gelöscht wird. Der `s3:ObjectTagging:Put`-Ereignistyp benachrichtigt Sie, wenn ein Tag auf einem Objekt PUTiert wird oder wenn ein vorhandener Tag aktualisiert wird. Der `s3:ObjectTagging:Delete`-Ereignistyp benachrichtigt Sie, wenn ein Tag aus einem Objekt entfernt wird. Weitere Informationen finden Sie unter [Aktivieren von Ereignisbenachrichtigungen](#).

Weitere Informationen über die Objektmarkierung finden Sie in den folgenden Themen:

Themen

- [Markierungs- und Zugriffskontrollrichtlinien](#)
- [Verwalten von Objekt-Markierungen](#)


Markierungs- und Zugriffskontrollrichtlinien

Sie können außerdem Berechtigungsrichtlinien (Bucket- und Benutzerrichtlinien) verwenden, um Berechtigungen für das Objekt-Tagging zu verwalten. Informationen über Richtlinienaktionen finden Sie in den folgenden Themen:

- [Objektoperationen](#)
- [Bucket-Operationen](#)

Objekt-Markierungen bieten eine differenzierte Zugriffskontrolle für die Verwaltung von Berechtigungen. Sie können bedingte Berechtigungen basierend auf Objekt-Markierungen erteilen. Amazon S3 unterstützt die folgenden Bedingungsschlüssel, die Sie verwenden können, um bedingte Berechtigungen basierend auf Objekt-Markierungen zu erteilen.

- `s3:ExistingObjectTag/<tag-key>` – Verwenden Sie diesen Bedingungsschlüssel, um zu überprüfen, ob ein vorhandenes Objekt-Tag den spezifischen Tag-Schlüssel und -Wert besitzt.

 Note

Wenn Sie Berechtigungen für die PUT Object- und DELETE Object-Operationen erteilen, wird dieser Bedingungsschlüssel nicht unterstützt. Dies bedeutet, dass Sie keine Richtlinie erstellen können, um einem Benutzer zu gestatten, ein Objekt basierend auf seinen vorhandenen Markierungen zu löschen oder zu überschreiben.

- `s3:RequestObjectTagKeys` – Verwenden Sie diesen Bedingungsschlüssel, um die Tag-Schlüssel einzuschränken, die Sie für Objekte zulassen wollen. Dies ist nützlich, wenn Sie Objekten mithilfe der Objektanforderungen PutObjectTagging PutObject AND und POST Tags hinzufügen.
- `s3:RequestObjectTag/<tag-key>` – Verwenden Sie diesen Bedingungsschlüssel, um die Tag-Schlüssel und -Werte einzuschränken, die Sie für Objekte zulassen wollen. Dies ist nützlich, wenn Sie Objekten mithilfe der Bucket-Anfragen PutObjectTagging PutObject AND und POST Tags hinzufügen.

Eine vollständige Liste der für den Amazon-S3-Service spezifischen Bedingungsschlüssel finden Sie unter [Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln](#). Die folgenden Berechtigungsrichtlinien zeigen, wie die Objektmarkierung eine differenzierte Zugriffsberechtigungsverwaltung ermöglicht.

Example 1: Einem Benutzer nur das Lesen von Objekten gestatten, die einen bestimmten Tag-Schlüssel und -Wert besitzen

Die folgende Berechtigungsrichtlinie beschränkt einen Benutzer darauf, nur Objekte zu lesen, die den Tag-Schlüssel und -Wert `environment: production` haben. Diese Richtlinie verwendet den `s3:ExistingObjectTag`-Bedingungsschlüssel, um den Tag-Schlüssel und -Wert anzugeben.

```
{
  "Version": "2012-10-17",
```

```

"Statement": [
{
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:role/JohnDoe"
    ]
  },
  "Effect": "Allow",
  "Action": ["s3:GetObject", "s3:GetObjectVersion"],
  "Resource": "arn:aws:s3::DOC-EXAMPLE-BUCKET/*",
  "Condition": {
    "StringEquals":
      {"s3:ExistingObjectTag/environment": "production"}
  }
}
]
}

```

Example 2: Einschränken, welche Objekt-Tag-Schlüssel Benutzer hinzufügen können

Die folgende Berechtigungsrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen Objekt Markierungen hinzuzufügen. Die Bedingung verwendet den Bedingungsschlüssel `s3:RequestObjectTagKeys`, um die zulässigen Tag-Schlüssel zu spezifizieren, z. B. `Owner` oder `CreationDate`. Weitere Informationen finden Sie unter [Erstellen einer Bedingung, die mehrere Schlüsselwerte testet](#) im IAM-Benutzerhandbuch.

Die Richtlinie stellt sicher, dass jeder in der Anfrage angegebene Tag-Schlüssel ein autorisierter Tag-Schlüssel ist. Der `ForAnyValue`-Qualifizierer in der Bedingung stellt sicher, dass mindestens einer der spezifizierten Schlüssel in der Anfrage enthalten ist.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":["
      "arn:aws:iam::111122223333:role/JohnDoe"
    ]
    }},
  "Effect": "Allow",
  "Action": [
    "s3:PutObjectTagging"
  ],

```

```

    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {"ForAnyValue:StringEquals": {"s3:RequestObjectTagKeys": [
      "Owner",
      "CreationDate"
    ]}
  }
}
]
}
}

```

Example 3: Einen spezifischen Tag-Schlüssel und -Wert verlangen, wenn Benutzern erlaubt wird, Objekt-Tags hinzuzufügen

Die folgende Beispielrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen Objekt Markierungen hinzuzufügen. Die Bedingung verlangt, dass der Benutzer einen spezifischen Tag-Schlüssel (z. B. *Project*) mit dem Wert *X* angibt.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
      "arn:aws:iam::111122223333:user/JohnDoe"
    ]},
    "Effect": "Allow",
    "Action": [
      "s3:PutObjectTagging"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"}
  }
}
]
}

```

Verwalten von Objekt-Markierungen

In diesem Abschnitt wird erklärt, wie Sie Objekt-Tags mithilfe der AWS SDKs für Java und .NET oder der Amazon S3 S3-Konsole verwalten können.

Das Markieren von Objekten ermöglicht Ihnen, Speicher zu kategorisieren. Jedes Tag ist ein Schlüssel-Wert-Paar, für das folgende Regeln gelten:

- Sie können einem Objekt bis zu 10 Markierungen zuordnen. Einem Objekt zugeordnete Markierungen müssen eindeutige Tag-Schlüssel haben.
- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Amazon-S3-Objekt-Tags werden intern in UTF-16 dargestellt. Beachten Sie, dass Zeichen in UTF-16 entweder 1 oder 2 Zeichenpositionen einnehmen.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#). Weitere Informationen zu Tag-Einschränkungen finden Sie unter [Einschränkungen benutzerdefinierter Markierungen](#) im AWS Billing and Cost Management Benutzerhandbuch.

Verwenden der S3-Konsole

Hinzufügen von Markern zu einem Objekt

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, denen Sie Markierungen hinzufügen möchten.

Sie können auch optional zu einem Ordner navigieren.

3. Aktivieren Sie in der Liste Objekte das Kontrollkästchen neben den Namen der Objekte, denen Sie Markierungen hinzufügen möchten.
4. Wählen Sie im Menü Aktionen die Option Markierungen bearbeiten.
5. Überprüfen Sie die aufgelisteten Objekte und wählen Sie Markierungen hinzufügen.
6. Jedes Objekt-Tag ist ein Schlüssel-Wert-Paar. Geben Sie einen Key (Schlüssel) und einen Value (Wert) ein. Um ein weiteres Tag hinzuzufügen, wählen Sie Add Tag (Tag hinzufügen).

Sie können bis zu 10 Marker für ein Objekt eingeben.

7. Wählen Sie Save Changes (Änderungen speichern).

Amazon S3 fügt die Markierungen zu den angegebenen Objekten hinzu.

Weitere Informationen finden Sie auch unter [Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole](#) und [Objekte hochladen](#) in diesem Handbuch.

Verwenden der AWS SDKs

Java

Das folgende Beispiel zeigt, wie Sie mithilfe von Tags für ein neues Objekt festlegen und Tags für ein vorhandenes Objekt abrufen oder ersetzen können. AWS SDK for Java Weitere Informationen über das Markieren von Objekten finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#). Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class ManagingObjectTags {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";
        String keyName = "**** Object key ****";
        String filePath = "**** File path ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
```

```
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(clientRegion)
        .build();

    // Create an object, add two new tags, and upload the object to Amazon
S3.
    PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
new File(filePath));
    List<Tag> tags = new ArrayList<Tag>();
    tags.add(new Tag("Tag 1", "This is tag 1"));
    tags.add(new Tag("Tag 2", "This is tag 2"));
    putRequest.setTagging(new ObjectTagging(tags));
    PutObjectResult putResult = s3Client.putObject(putRequest);

    // Retrieve the object's tags.
    GetObjectTaggingRequest getTaggingRequest = new
GetObjectTaggingRequest(bucketName, keyName);
    GetObjectTaggingResult getTagsResult =
s3Client.getObjectTagging(getTaggingRequest);

    // Replace the object's tags with two new tags.
    List<Tag> newTags = new ArrayList<Tag>();
    newTags.add(new Tag("Tag 3", "This is tag 3"));
    newTags.add(new Tag("Tag 4", "This is tag 4"));
    s3Client.setObjectTagging(new SetObjectTaggingRequest(bucketName,
keyName, new ObjectTagging(newTags)));
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Das folgende Beispiel zeigt, wie Sie mithilfe von AWS SDK for .NET die Tags für ein neues Objekt festlegen und die Tags für ein vorhandenes Objekt abrufen oder ersetzen können. Weitere

Informationen über das Markieren von Objekten finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Getting Started with the AWS SDK for .NET](#) im AWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    public class ObjectTagsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for the new object ***";
        private const string filePath = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            PutObjectWithTagsTestAsync().Wait();
        }

        static async Task PutObjectWithTagsTestAsync()
        {
            try
            {
                // 1. Put an object with tags.
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    FilePath = filePath,
                    TagSet = new List<Tag>{
                        new Tag { Key = "Keyx1", Value = "Value1"},

```



```
        new Tag { Key = "Keyx2", Value = "Value2" }
    }
};

PutObjectResponse response = await
client.PutObjectAsync(putRequest);
// 2. Retrieve the object's tags.
GetObjectTaggingRequest getTagsRequest = new GetObjectTaggingRequest
{
    BucketName = bucketName,
    Key = keyName
};

GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);
for (int i = 0; i < objectTags.Tagging.Count; i++)
    Console.WriteLine("Key: {0}, Value: {1}",
objectTags.Tagging[i].Key, objectTags.Tagging[i].Value);

// 3. Replace the tagset.

Tagging newTagSet = new Tagging();
newTagSet.TagSet = new List<Tag>{
    new Tag { Key = "Key3", Value = "Value3"},
    new Tag { Key = "Key4", Value = "Value4" }
};

PutObjectTaggingRequest putObjTagsRequest = new
PutObjectTaggingRequest()
{
    BucketName = bucketName,
    Key = keyName,
    Tagging = newTagSet
};

PutObjectTaggingResponse response2 = await
client.PutObjectTaggingAsync(putObjTagsRequest);

// 4. Retrieve the object's tags.
GetObjectTaggingRequest getTagsRequest2 = new
GetObjectTaggingRequest();
getTagsRequest2.BucketName = bucketName;
getTagsRequest2.Key = keyName;
```

```
        GetObjectTaggingResponse objectTags2 = await
client.GetObjectTaggingAsync(getTagsRequest2);
        for (int i = 0; i < objectTags2.Tagging.Count; i++)
            Console.WriteLine("Key: {0}, Value: {1}",
objectTags2.Tagging[i].Key, objectTags2.Tagging[i].Value);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine(
            "Error encountered ***. Message:'{0}' when writing an
object"
                , e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine(
            "Encountered an error. Message:'{0}' when writing an object"
                , e.Message);
    }
}
}
```

Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets

Um die Speicherkosten oder andere Kriterien für einzelne Projekte oder Gruppen von Projekten nachzuverfolgen, kennzeichnen Sie Ihre Amazon S3-Buckets unter Verwendung der Kostenzuordnungs-Markierungen. Ein Kostenzuordnungs-Tag ist ein Schlüssel-Wert-Paar, das Sie definieren und mit einem S3-Bucket verknüpfen. Nachdem Sie Kostenzuordnungs-Markierungen aktiviert haben, verwendet AWS die Markierungen, um Ihre Ressourcenkosten in Ihrem Kostenzuordnungsbericht zu gruppieren. Kostenzuordnungs-Markierungen können nur zur Kennzeichnung von Buckets verwendet werden. Weitere Informationen zu Markierungen, die zur Kennzeichnung von Objekten verwendet werden, finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

Der Kostenzuordnungsbericht listet die AWS -Nutzung für Ihr Konto nach Produktkategorie und verbundenem Kontobenutzer auf. Der Bericht enthält die gleichen Einzelposten wie der detaillierte Fakturierungsbericht (siehe [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#)), und zusätzliche Spalten für Ihre Tag-Schlüssel.

AWS bietet zwei Arten von Kostenzuordnungs-Tags, einen AWS-generierten Tag und benutzerdefinierte Tags. AWS definiert, erstellt und wendet den AWS-generierten `createdBy`-Tag nach einem Amazon S3-CreateBucket-Ereignis für Sie an. Sie definieren, erstellen und verwenden benutzerdefinierte Markierungen für Ihren S3-Bucket.

Sie müssen beide Tag-Typen in der Fakturierungs- und Kostenverwaltungskonsole separat aktivieren, damit sie in Ihren monatlichen Berichten angezeigt werden können. Weitere Informationen über von AWS generierte Markierungen finden Sie unter [Von AWS generierte Kostenzuordnungs-Markierungen](#).

- Informationen zum Erstellen von Markierungen in der Konsole finden Sie unter [Anzeigen der Eigenschaften eines S3-Buckets](#).
- Informationen zum Erstellen von Markierungen mit der Amazon S3-API finden Sie unter [PUT Bucket Tagging](#) in der Amazon Simple Storage Service API-Referenz.
- Informationen zum Erstellen von Markierungen mit der AWS CLI finden Sie unter [put-bucket-tagging](#) in der AWS CLI-Befehlsreferenz.
- Weitere Informationen zur Aktivierung von Markierungen finden Sie unter [Using cost allocation tags \(Verwendung von Kostenzuordnungs-Markierungen\)](#) im AWS Billing-Benutzerhandbuch.

Benutzerdefinierte Kostenzuordnungs-Markierungen

Ein benutzerdefiniertes Kostenzuordnungs-Tag umfasst folgende Komponenten:

- Der Tag-Schlüssel. Der Tag-Schlüssel ist der Name der Markierungen. Im Tag `project/Trinity` beispielsweise ist `project` der Schlüssel. Der Tag-Schlüssel ist eine Zeichenfolge, in der zwischen Groß- und Kleinschreibung unterschieden wird und die 1 bis 128 Unicode-Zeichen enthalten kann.
- Der Tag-Wert. Der Tag-Wert ist eine erforderliche Zeichenfolge. Im Tag `project/Trinity` beispielsweise ist `Trinity` der Wert. Der Tag-Wert ist eine Zeichenfolge, in der zwischen Groß- und Kleinschreibung unterschieden wird und die 0 bis 256 Unicode-Zeichen enthalten kann.

Weitere Informationen zu den zulässigen Zeichen für benutzerdefinierte Markierungen und anderen Einschränkungen finden Sie unter [User-Defined Tag Restrictions \(Einschränkungen benutzerdefinierter Markierungen\)](#) im AWS Billing-Benutzerhandbuch. Weitere Informationen über benutzerdefinierte Markierungen finden Sie unter [User-Defined Cost Allocation Markierungen \(Benutzerdefinierte Kostenzuordnungs-Markierungen\)](#) im AWS Billing-Benutzerhandbuch.

S3-Bucket-Markierungen

Jeder S3-Bucket verfügt über einen Tag-Satz. Ein Tag-Satz enthält alle Markierungen, die diesem Bucket zugewiesen sind. Ein Tag-Satz kann bis zu 50 Markierungen enthalten oder leer sein. Schlüssel müssen eindeutig innerhalb eines Tag-Satzes sein, aber die Werte in einem Tag-Satz müssen nicht eindeutig sein. Sie können beispielsweise denselben Wert in Tag-Sätzen mit den Namen project/Trinity und cost-center/Trinity haben.

Wenn Sie in einem Bucket ein Tag mit demselben Schlüssel wie für ein vorhandenes Tag hinzufügen, wird der alte Wert mit dem neuen überschrieben.

AWS ordnet Ihren Markierungen keine semantische Bedeutung zu. Wir interpretieren Markierungen streng als Zeichenfolgen.

Um Markierungen hinzuzufügen, zu bearbeiten oder zu löschen, verwenden Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI) oder die Amazon S3-API.

Weitere Infos

- [Verwenden von Kostenzuordnungs-Markierungen](#) im AWS Billing-Benutzerhandbuch.
- [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#)
- [AWS Billing Berichte für Amazon S3](#)

Fakturierung und Nutzungsberichte für Amazon S3

Wenn Sie Amazon S3 verwenden, müssen Sie keine Vorabgebühren zahlen oder sich darauf festlegen, wie viele Inhalte Sie speichern werden. Wie bei anderen AWS-Services Unternehmen zahlen Sie nutzungsabhängig und zahlen nur für das, was Sie tatsächlich nutzen.

AWS bietet die folgenden Berichte für Amazon S3:

- Abrechnungsberichte — Mehrere Berichte, die einen Überblick über alle Aktivitäten bieten AWS-Services, die Sie verwenden, einschließlich Amazon S3. AWS stellt dem Besitzer des S3-Buckets immer die Amazon S3 S3-Gebühren in Rechnung, es sei denn, der Bucket wurde als Bucket mit anfordernder Zahlung erstellt. Weitere Informationen über Zahlung durch den Anforderer finden Sie unter [Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung](#). Weitere Informationen über Fakturierungsberichte finden Sie unter [AWS Billing Berichte für Amazon S3](#).

- Nutzungsbericht – Eine Zusammenfassung von Aktivitäten für einen bestimmten Service, zusammengefasst nach Stunde, Tag und Monat. Sie können wählen, welche Nutzungsart und welche Operation aufgenommen werden soll. Sie können auch festlegen, wie die Daten aggregiert werden. Weitere Informationen finden Sie unter [AWS Nutzungsbericht für Amazon S3](#).

Die folgenden Themen enthalten Informationen zur Fakturierung und zu Nutzungsberichten für Amazon S3.

Themen

- [AWS Billing Berichte für Amazon S3](#)
- [AWS Nutzungsbericht für Amazon S3](#)
- [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#)
- [Abrechnung für Amazon S3 S3-Fehlerantworten](#)

AWS Billing Berichte für Amazon S3

Ihre monatliche Rechnung von AWS trennt Ihre Nutzungsinformationen und Kosten nach AWS-Service Funktion. Es sind mehrere AWS Billing Berichte verfügbar: der Monatsbericht, der Kostenzuweisungsbericht und detaillierte Abrechnungsberichte. Informationen zum Anzeigen Ihrer Fakturierungsberichte finden Sie unter [Viewing Your Bill \(Anzeigen Ihrer Rechnung\)](#) im AWS Billing - Benutzerhandbuch.

Um Ihre AWS Nutzung zu verfolgen und die geschätzten Gebühren für Ihr Konto anzugeben, können Sie Folgendes einrichten AWS Cost and Usage Reports. Weitere Informationen finden Sie unter [Was sind AWS Cost and Usage Reports?](#) im Leitfaden AWS für Datenexporte.

Sie können auch einen Nutzungsbericht herunterladen, der Ihnen weitere Informationen über Ihre Amazon-S3-Nutzung mitteilt, als die Fakturierungs-Berichte. Weitere Informationen finden Sie unter [AWS Nutzungsbericht für Amazon S3](#).

Die folgende Tabelle listet die Gebühren für die Amazon-S3-Nutzung auf.

Amazon-S3-Nutzungsgebühren

Gebühr	Kommentare
Speicher	Sie zahlen für das Speichern von Objekten in S3-Buckets. Der Tarif, der Ihnen in Rechnung

Gebühr	Kommentare
	<p>gestellt wird, hängt von der Größe Ihrer Objekte, der Aufbewahrungsdauer der Objekte im Laufe des Monats und der Speicherklasse ab. Amazon S3 bietet die folgenden Speicherlassen: S3 Standard, S3 Express One Zone, S3 Intelligent-Tiering, S3 Standard-IA (IA für seltenen Zugriff), S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive oder Reduced Redundancy Storage (RRS). Weitere Informationen über Speicherklassen finden Sie unter Verwenden von Amazon-S3-Speicherklassen.</p> <p>Beachten Sie, dass Ihnen bei aktivierter S3-Versionierung jede Version eines Objekts, die aufbewahrt wird, in Rechnung gestellt wird. Weitere Informationen über das Versioning finden Sie unter Wie funktioniert S3-Versioning.</p>
Überwachen und Automatisieren	Für jedes in der Speicherklasse S3 Intelligent-Tiering gespeicherte Objekt zahlen Sie eine monatliche Überwachungs- und Automatisierungsgebühr zur Überwachung der Zugriffsmuster und Verschiebung der Objekte zwischen den Zugriffsstufen des S3 Intelligent-Tiering.
Anforderungen	Sie zahlen für Anfragen, z. B. GET Anfragen, die für Ihre S3-Buckets und -Objekte gestellt werden. Dazu zählen auch Lebenszyklus-Anfragen. Die Gebühren für Anfragen hängen davon ab, welche Art von Anfrage Sie stellen. Weitere Informationen zu den Anfragepreisen erhalten Sie unter Amazon-S3-Preise .

Gebühr	Kommentare
Abrufe	Sie zahlen für das Abrufen von Objekten, die sich im Speicher S3 Standard-IA, S3 One Zone-IA-, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive befinden.
Vorzeitige Löschungen	Wenn Sie ein Objekt im Speicher S3 Standard-IA, S3 One Zone-IA, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive löschen, bevor die minimale Speicherzusage abgelaufen ist, zahlen Sie eine Gebühr für das vorzeitige Löschen für dieses Objekt.
Speicherverwaltung	Sie zahlen für die Speicherverwaltungsfunktionen (Amazon S3 S3-Inventar, Analytik und Objekt-Tagging), die in den Buckets Ihres Kontos aktiviert sind.
Bandbreite	<p>Sie zahlen für die gesamte Bandbreite in und aus Amazon S3 mit Ausnahme des Folgenden:</p> <ul style="list-style-type: none">• Datenübertragung aus dem Internet• Daten, die an eine Amazon Elastic Compute Cloud (AmazonEC2) -Instance übertragen werden, wenn sich die Instance im selben AWS-Region wie der S3-Bucket befindet• An Amazon übertragene Daten CloudFront (CloudFront) <p>Sie zahlen auch eine Gebühr für alle Daten, die mit Amazon S3 Transfer Acceleration übertragen werden.</p>

Detaillierte Informationen zu den Amazon S3-Nutzungsgebühren für Speicher, Datenübertragung und Services finden Sie unter [Amazon S3 S3-Preise](#) und [Amazon S3 FAQs](#).

Informationen zum Verständnis der in den Abrechnungs- und Nutzungsberichten für Amazon S3 verwendeten Codes und Abkürzungen finden Sie unter [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#).

Weitere Informationen

- [AWS Nutzungsbericht für Amazon S3](#)
- [Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets](#)
- [AWS Billing und Kostenmanagement](#)
- [Amazon S3 – Preise](#)

AWS Nutzungsbericht für Amazon S3

Wenn Sie einen Nutzungsbericht herunterladen, können Sie die Nutzungsdaten nach Stunde, Tag und Monat zusammenfassen. Der Amazon S3 S3-Nutzungsbericht listet die Vorgänge nach Nutzungstyp und auf AWS-Region. Für detailliertere Nutzungsberichte über Ihre Amazon-S3-Speichernutzung laden Sie dynamisch generierte AWS -Nutzungsberichte herunter. Sie können wählen, welche Nutzungsart, welche Operation und welcher Zeitraum aufgenommen werden soll. Sie können auch festlegen, wie die Daten aggregiert werden. Weitere Informationen zu Nutzungsberichten finden Sie unter [AWS Nutzungsbericht](#) im AWS Data Exports User Guide.

Der Amazon-S3-Nutzungsbericht enthält die folgenden Informationen:

- Service – Amazon S3
- Operation – Die Operation, die für Ihren Bucket oder Ihr Objekt durchgeführt wird. Eine detaillierte Erklärung der Amazon-S3-Vorgänge finden Sie unter [Nachverfolgen von Vorgängen in Ihren Nutzungsberichten](#).
- UsageType— Einer der folgenden Werte:
 - Ein Code zur Identifizierung des Speichertyps
 - Ein Code zur Identifizierung der Anfrage
 - Ein Code zur Identifizierung des Abruftyps
 - Ein Code zur Identifizierung des Typs der Datenübertragung

- Ein Code, der vorzeitiges Löschen aus dem S3 Intelligent-Tiering, S3 Standard-IA-, S3-One-Zone-Infrequent-Access (S3 One Zone-IA)-, S3 Glacier Flexible Retrieval- oder S3 Glacier Deep Archive Speicher identifiziert
- StorageObjectCount – Die Anzahl der Objekte in einem bestimmten Bucket

Eine detaillierte Erklärung der Amazon-S3-Nutzungstypen finden Sie unter [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#).

- Resource (Ressource) – Der Name des Buckets, der der aufgelisteten Nutzung zugeordnet ist.
- StartTime— Startzeit des Tages, für den die Verwendung gilt, in koordinierter Weltzeit (UTC).
- EndTime— Endzeit des Tages, für den die Nutzung gilt, in koordinierter Weltzeit (UTC).
- UsageValue— Einer der folgenden Volumenwerte. Die typische Maßeinheit für Daten ist Gigabyte (GB). Abhängig vom Dienst und dem Bericht können stattdessen Terabytes (TB) angezeigt werden.
 - Die Anzahl der Anforderungen in dem angegebenen Zeitraum
 - Die übertragene Datenmenge
 - Die Menge der in einer bestimmten Stunde gespeicherten Daten
 - Die Menge der Daten im Zusammenhang mit Wiederherstellungen aus dem Speicher S3 Standard-IA, S3 One Zone-IA, S3 Glacier Flexible Retrieval-, oder S3 Glacier Deep Archive

 Tip

Detaillierte Informationen über alle Anfragen, die Amazon S3 für Ihre Objekte empfängt, aktivieren Sie die Server-Zugriffsprotokollierung für Ihre Buckets Weitere Informationen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

Sie können einen Nutzungsbericht als Datei XML oder als Datei mit kommagetrennten Werten (CSV) herunterladen. Im Folgenden finden Sie ein Beispiel für einen CSV Nutzungsbericht, der in einer Tabellenkalkulationsanwendung geöffnet wurde.

Service	Operation	UsageType	Resource	StartTime	EndTime	UsageValue
AmazonS3	HeadBucket	USW2-C3DataTransfer-Out-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	15309
AmazonS3	PutObject	USW2-C3DataTransfer-In-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	19062
AmazonS3	HeadBucket	USW2-Requests-Tier2	admin-created3	6/1/2017 0:00	7/1/2017 0:00	68
AmazonS3	PutObjectForRepl	USW1-Requests-SIA-Tier1	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	178294
AmazonS3	PutObjectForRepl	USW1-USW2-AWS-In-Bytes	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	387929083
AmazonS3	GetObjectForRepl	USW2-Requests-NoCharge	admin-created3	6/1/2017 0:00	7/1/2017 0:00	108
AmazonS3	GetObjectForRepl	USW2-USW1-AWS-Out-Bytes	my-test-bucket-bash	6/1/2017 0:00	7/1/2017 0:00	387910021

Weitere Informationen finden Sie unter [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#).

Der AWS Nutzungsbericht wird heruntergeladen

Sie können einen Nutzungsbericht als CSV Datei XML oder herunterladen.

Den Nutzungsbericht herunterladen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Titelleiste Ihren Benutzernamen oder Ihre Konto-ID und dann Billing and Cost Management aus.
3. Wählen Sie im Navigationsbereich die Option Kosten- und Nutzungsberichte aus.
4. Wählen Sie unter AWS Nutzungsbericht die Option Nutzungsbericht erstellen aus.
5. Wählen Sie auf der Seite Nutzungsbericht herunterladen die folgenden Einstellungen aus:
 - Services — Wählen Sie Amazon Simple Storage Service.
 - Usage Types (Nutzungstypen) – Eine detaillierte Erläuterung der Amazon-S3-Nutzungstypen finden Sie unter [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#).
 - Operation – Eine detaillierte Erläuterung der Amazon-S3-Vorgänge finden Sie unter [Nachverfolgen von Vorgängen in Ihren Nutzungsberichten](#).
 - Time Period (Zeitraum) – Der Zeitraum, der in dem Bericht erfasst werden soll.
 - Report Granularity (Berichtsgranularität) – Gibt an, ob in dem Bericht Zwischensummen nach Stunde, Tag oder Monat angezeigt werden sollen.
6. Wählen Sie Herunterladen, wählen Sie das Download-Format (XMLBericht oder CSVBericht) und folgen Sie dann den Anweisungen, um den Bericht zu öffnen oder zu speichern.

Weitere Informationen

- [Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3](#)
- [AWS Billing Berichte für Amazon S3](#)

Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3

Amazon-S3-Fakturierungs- und Nutzungsberichte verwenden Codes und Abkürzungen. Für Nutzungstypen in der folgenden Tabelle ersetzen Sie *regionregion1*, und *region2* durch Abkürzungen aus dieser Liste:

- APE1: Asien-Pazifik (Hongkong)
- APN1: Asien-Pazifik (Tokio)
- APN2: Asien-Pazifik (Seoul)
- APN3: Asien-Pazifik (Osaka)
- APS1: Asien-Pazifik (Singapur)
- APS2: Asien-Pazifik (Sydney)
- APS3: Asien-Pazifik (Mumbai)
- APS4: Asien-Pazifik (Jakarta)
- APS5: Asien-Pazifik (Hyderabad)
- APS6: Asien-Pazifik (Melbourne)
- CAN1: Kanada (Zentral)
- CAN2: Kanada West (Calgary)
- CNN1: China (Peking)
- CNW1: China (Ningxia)
- AFS1: Afrika (Kapstadt)
- EUC2: Europa (Zürich)
- EUN1: Europa (Stockholm)
- EUS2: Europa (Spanien)
- EUC1: Europa (Frankfurt)

- EU: Europa (Irland)
- EUS1: Europa (Mailand)
- EUW2: Europa (London)
- EUW3: Europa (Paris)
- ILC1: Israel (Tel Aviv)
- MEC1: Naher Osten (UAE)
- MES1: Naher Osten (Bahrain)
- SAE1: Südamerika (São Paulo)
- UGW1: AWS GovCloud (US-West)
- UGE1: AWS GovCloud (US-Ost)
- USE1(oder kein Präfix): US East (Nord-Virginia)
- USE2: USA Ost (Ohio)
- USW1: USA West (Nordkalifornien)
- USW2: USA West (Oregon)

Für die Verwendungsarten von S3 Multiregion Access Points in der folgenden Tabelle ersetzen Sie *regiongroup1* und *regiongroup2* durch Abkürzungen aus dieser Liste:

- AP: Asien-Pazifik
- AU: Australien
- EU: Europa
- IN: Indien
- NA: Nordamerika
- SA: Südamerika

Bei Regionalgruppen handelt es sich um geografische Gruppierungen von mehreren AWS-Regionen. Weitere Informationen finden Sie unter [Regionen und Availability Zones](#). Weitere Informationen zu den Preisen nach AWS-Region finden Sie unter [Amazon S3 – Preise](#).

Die erste Spalte in der folgenden Tabelle listet die Nutzungstypen auf, die in Ihren Fakturierungs- und Nutzungsberichten vorkommen. Die typische Maßeinheit für Daten ist Gigabyte (GB). Abhängig vom Dienst und dem Bericht können stattdessen Terabytes (TB) angezeigt werden.

Verwendungstypen

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region1-region2</i> -AWS-In-A Bytes	GB	Stundensatz	Die Menge der beschleunigten Daten, die von <i>region1</i> übertragen werden <i>region2</i>
<i>region1-region2</i> -AWS-In-A Bytes-T1	GB	Stundensatz	Die Menge der beschleunigten T1-Daten <i>region2</i> , <i>region1</i> von der aus übertragen wurde, wobei T1 sich auf CloudFront Anfragen an Präsenzstellen (POPs) in den Vereinigten Staaten, Europa und Japan bezieht
<i>region1-region2</i> -AWS-In-A Bytes-T2	GB	Stundensatz	Die Menge der beschleunigten T2-Daten, von der <i>region1</i> aus übertragen wurde <i>region2</i> , wobei T2 sich auf CloudFront Anfragen POPs an alle anderen AWS Edge-Standorte bezieht
<i>region1-region2</i> -AWS-In-Bytes	GB	Stundensatz	Die Menge der übertragenen Daten <i>region1</i> von <i>region2</i>
<i>region1-region2</i> -AWS-Out-Bytes	GB	Stundensatz	Die Menge der beschleunigten Daten, die von <i>region1</i> bis übertragen wurden <i>region2</i>

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region1-region2</i> -AWS-Out-Bytes-T1	GB	Stundensatz	Die Menge der beschleunigten T1-Daten <i>region2</i> , <i>region1</i> die von bis übertragen wurden, wobei sich T1 auf CloudFront Anfragen POPs in den Vereinigten Staaten, Europa und Japan bezieht
<i>region1-region2</i> -AWS-Out-Bytes-T2	GB	Stundensatz	Die Menge der beschleunigten T2-Daten, die von <i>region1</i> bis übertragen wurden <i>region2</i> , wobei T2 sich auf CloudFront Anfragen POPs an alle anderen AWS Edge-Standorte bezieht
<i>region1-region2</i> -AWS-Out-Bytes	GB	Stundensatz	Die Menge der Daten, die von <i>region1</i> bis übertragen wurden <i>region2</i>
<i>region</i> -BatchOperations-Jobs	Anzahl	Stundensatz	Die Anzahl der ausgeführten S3-Batchvorgangsaufträge
<i>region</i> -BatchOperations-Objects	Anzahl	Stundensatz	Die Anzahl der von S3-Batchvorgänge ausgeführten Objektvorgänge

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Bulk-Retrieval-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Massenanforderungen von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive abgerufen wurden
<i>region</i> -BytesDeleted-GDA	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Glacier Deep Archive Archive-Speicher gelöscht wurde
<i>region</i> -BytesDeleted-GIR	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Glacier Instant Retrieval-Speicher gelöscht wurde.
<i>region</i> -BytesDeleted-GLACIER	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Glacier Flexible Retrieval-Speicher gelöscht wurde
<i>region</i> -BytesDeleted-INT	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Intelligent-Tiering-Speicher gelöscht wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -BytesDeleted-RRS	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem Speicher mit reduziertem Redundanzspeicher () gelöscht wurde RRS
<i>region</i> -BytesDeleted-SIA	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3-Standard-IA-Speicher gelöscht wurde
<i>region</i> -BytesDeleted-STANDARD	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3-Standardspeicher gelöscht wurde
<i>region</i> -BytesDeleted-ZIA	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 One Zone-IA-Speicher gelöscht wurde
<i>region</i> -C3DataTransfer-In-Bytes	GB	Stundensatz	Die Datenmenge, die EC2 innerhalb desselben Zeitraums von Amazon nach Amazon S3 übertragen wurde AWS-Region

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -C3DataTransfer-Out-Bytes	GB	Stundensatz	Die Datenmenge, die EC2 innerhalb desselben Zeitraums von Amazon S3 zu Amazon übertragene wurde AWS-Region
<i>region</i> -CloudFront-In-Bytes	GB	Stundensatz	Die Datenmenge, die in eine oder AWS-Region aus einer CloudFront Distribution übertragen wurde
<i>region</i> -CloudFront-Out-Bytes	GB	Stundensatz	Die Datenmenge, die von einer AWS-Region zu einer CloudFront Distribution übertragen wurde
<i>region</i> -DataTransfer-In-Bytes	GB	Stundensatz	Die Menge der aus dem Internet zu Amazon S3 übertragenen Daten
<i>region</i> -DataTransfer-Out-Bytes	GB	Stundensatz	Die Menge der von Amazon S3 ins Internet übertragenen Daten ¹
<i>region</i> -DataTransfer-Regional-Bytes	GB	Stundensatz	Die Datenmenge, die von Amazon S3 auf AWS Ressourcen innerhalb desselben übertragen wurde AWS-Region

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -EarlyDelete-ByteHrs	GB-Stunden	Stündlich	Anteilige Speichernutzung für Objekte, die aus dem Speicher S3 Glacier Flexible Retrieval gelöscht wurden, bevor die 90-tägige Mindestnutzung endete ²
<i>region</i> -EarlyDelete-GDA	GB-Stunden	Stundensatz	Anteilige Speichernutzung für Objekte, die aus dem S3-Glacier-Deep-Archive-Speicher gelöscht wurden, bevor die 180-tägige Mindestnutzung endete ²
<i>region</i> -EarlyDelete-GIR	GB-Stunden	Stündlich	Anteilige Speichernutzung für Objekte, die aus dem Speicher S3 Glacier Instant Retrieval gelöscht wurden, bevor die 90-tägige Mindestnutzung endete.
<i>region</i> -EarlyDelete-GIR-SmallObjects	GB-Stunden	Stündlich	Anteilige Speichernutzung für kleine Objekte (kleiner als 128 KB), die aus S3 Glacier Instant Retrieval gelöscht wurden, bevor die 90-tägige Mindestnutzung endete.

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -EarlyDelete-SIA	GB-Stunden	Stundensatz	Anteilige Speichernutzung für Objekte, die aus dem S3-Standard-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endet ³
<i>region</i> -EarlyDelete-SIA-SmallObjects	GB-Stunden	Stundensatz	Anteilige Speichernutzung für kleine Objekte (kleiner als 128 KB), die aus dem S3-Standard-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endet ³
<i>region</i> -EarlyDelete-ZIA	GB-Stunden	Stundensatz	Anteilige Speichernutzung für Objekte, die aus dem S3-One-Zone-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endet ³
<i>region</i> -EarlyDelete-ZIA-SmallObjects	GB-Stunden	Stundensatz	Anteilige Speichernutzung für kleine Objekte (kleiner als 128 KB), die aus dem S3-One-Zone-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endet ³

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Expedited-Retrieval-Bytes	GB	Stundensatz	Die Menge der Daten, die mit beschleunigten Anforderungen von S3 Glacier Flexible Retrieval abgerufen wurden
<i>region</i> -Inventory-Objects Listed	Objekte	Stundensatz	Die Anzahl der Objekte, die für eine Objektgruppe (Objekte werden nach Bucket oder Präfix gruppiert) in einer Lagerbestandsliste aufgelistet werden
<i>region</i> -Monitoring-Automation-INT	Objekte	Stundensatz	Die Anzahl an eindeutigen Objekten, die in der S3-Intelligent-Tiering-Speicherklasse überwacht und automatisch abgestuft werden
<i>region</i> -MRAP-Out-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points aus Buckets in einer Region übertragen wird (Preise für MRAP Datenweiterleitung).

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -MRAP-In-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points aus Buckets in einer Region übertragen wird (Preise für MRAP Datenrouting).
<i>regiongroup1-regiongroup2</i> -MRAP-Out-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points von einem Bucket <i>regiongroup1</i> zu einem Client <i>regiongroup2</i> außerhalb des Netzwerks übertragen wird. AWS
<i>regiongroup1-regiongroup2</i> -MRAP-In-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points <i>regiongroup1</i> von einem Client <i>regiongroup2</i> außerhalb des Netzwerks zu einem Bucket übertragen wird. AWS.
<i>region</i> -OverwriteBytes-Copy-GDA	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Glacier Deep Archive Archive-Speicher überschrieben wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -OverwriteBytes-Copy-GIR	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Glacier Instant Retrieval-Speicher überschrieben wurde.
<i>region</i> -OverwriteBytes-Copy-GLACIER	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Glacier Flexible Retrieval-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Copy-INT	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Intelligent-Tiering-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Copy-RRS	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem Speicher mit reduziertem Redundanzspeicher (1) überschrieben wurde RRS
<i>region</i> -OverwriteBytes-Copy-SIA	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3-Standard-IA-Speicher überschrieben wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -OverwriteBytes-Copy-STANDARD	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Standard-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Copy-ZIA	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 One Zone-IA-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Put-GDA	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Glacier Deep Archive Archive-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Put-GIR	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Glacier Instant Retrieval -Speicher überschrieben wurde.
<i>region</i> -OverwriteBytes-Put-GLACIER	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Glacier Flexible Retrieval -Speicher überschrieben wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -OverwriteBytes-Put-INT	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Intelligent-Tiering-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Put-RRS	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem Speicher mit reduziertem Redundanzspeicher (1) überschrieben wurde
<i>region</i> -OverwriteBytes-Put-SIA	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3-Standard-IA-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Put-STANDARD	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Standard-Speicher überschrieben wurde
<i>region</i> -OverwriteBytes-Put-ZIA	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 One Zone-IA-Speicher überschrieben wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region1-region2</i> -S3RTC-In-Bytes	GB	Monatlich	Die Datenmenge, die für S3 Replication Time Control (S3RTC) von bis <i>region2</i> nach den <i>region1</i> Vorgängen PutObject ReplTime ,, GetObjectReplTime InitiateMultipartUploadReplTime UploadPartReplTime , CompleteMultipartUploadReplTime und übertragen wurde WriteACLReplTime
<i>region1-region2</i> -S3RTC-Out-Bytes	GB	Monatlich	Die für S3 Replication Time Control (S3RTC) von bis <i>region1</i> nach <i>region2</i> den WriteACLReplTime Vorgängen PutObject ReplTime ,, GetObjectReplTime InitiateMultipartUploadReplTime UploadPartReplTime CompleteMultipartUploadReplTime , und übertragene Datenmenge

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Requests-GDA-Tier1	Anzahl	Stündlich	Die Anzahl von PUT, COPY, POST CreateMultipartUpload UploadPart , oder CompleteMultipartUpload Anfragen für S3 Glacier Deep Archive Archive-Objekte ⁶
<i>region</i> -Requests-GDA-Tier2	Anzahl	Stündlich	Die Anzahl von GET und HEAD Anfragen für S3 Glacier Deep Archive Archive-Objekte
<i>region</i> -Requests-GDA-Tier3	Anzahl	Stundensatz	Die Anzahl der S3 Glacier Deep Archive-Standard-Wiederherstellungsanforderungen
<i>region</i> -Requests-GDA-Tier5	Anzahl	Stundensatz	Die Anzahl der S3 Glacier Deep Archive-Wiederherstellungsmassenanforderungen
<i>region</i> -Requests-GIR-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST -Anfragen für S3 Glacier Instant Retrieval-Objekte.
<i>region</i> -Requests-GIR-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu S3 Glacier Instant Retrieval-Tier1 gehören, für S3 Glacier Instant Retrieval-Objekte.

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Requests-GLACIER-Tier1	Anzahl	Stündlich	Die Anzahl von PUT, COPYPOST, oder CompleteMultipartUpload Anfragen für S3 CreateMultipartUpload, Glacier UploadPart, Flexible Retrieval-Objekte
<i>region</i> -Requests-GLACIER-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht in S3 Glacier Flexible Retrieval-Objekten aufgeführt sind
<i>region</i> -Requests-INT-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST Anfragen für S3 Intelligent-Tiering-Objekte
<i>region</i> -Requests-INT-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu Stufe 1 gehören, für S3 Intelligent-Tiering-Objekte
<i>region</i> -Requests-SIA-Tier1	Anzahl	Stündlich	Die Anzahl von PUT, oder POST -Anfragen für S3 COPY Standard-IA-Objekte
<i>region</i> -Requests-SIA-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Nicht-S3 Glacier Instant Retrieval-Tier1-Anfragen für S3 Standard-IA-Objekte

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Requests-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST Anfragen für S3 Standard und Tags sowie Anfragen für RRS alle Buckets und Objekte LIST
<i>region</i> -Requests-Tier2	Anzahl	Stündlich	Die Anzahl der GET und aller anderen Anfragen, die nicht zu Stufe 1 gehören
<i>region</i> -Requests-Tier3	Anzahl	Stündlich	Die Anzahl der Lebenszyklusanforderungen an Wiederherstellungsanforderungen von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive und Standard S3 Glacier Flexible Retrieval
<i>region</i> -Requests-Tier4	Anzahl	Stündlich	Die Anzahl der Lebenszyklus-Übergänge in Speicher von S3 Glacier Instant Flexible Retrieval, S3 Intelligent-Tiering, S3 Standard-IA, oder S3 One Zone-IA
<i>region</i> -Requests-Tier5	Anzahl	Stündlich	Die Anzahl der Wiederherstellungsmassenanforderungen von S3 Glacier Flexible Retrieval

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Requests-Tier6	Anzahl	Stündlich	Die Anzahl der beschleunigten Wiederherstellungsanforderungen von S3 Glacier Flexible Retrieval
<i>region</i> -Requests-Tier8	Anzahl	Stündlich	Die Anzahl der S3 Access Grants-Anfragen
<i>region</i> -Requests-XZ-Tier1	Anzahl	Stündlich	Die Anzahl der PUT COPY Or-Anfragen für S3 Express One Zone-Objekte
<i>region</i> -Requests-XZ-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu S3 Express One Zone-Tier1 gehören, für S3 Express One Zone-Objekte
<i>region</i> -Requests-ZIA-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST Anfragen für S3 One Zone-IA-Objekte
<i>region</i> -Requests-ZIA-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu S3 One Zone-IA-Tier1 gehören, für S3 One Zone-IA-Objekte
<i>region</i> -Retrieval-GIR	GB	Stundensatz	Die Menge der Daten, die aus dem Speicher S3 Glacier Instant Retrieval abgerufen wurden.

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Retrieval-SIA	GB	Stundensatz	Die Menge der Daten, die aus dem S3-Standard-IA-Speicher abgerufen wurden
<i>region</i> -Retrieval-XZ	GB	Stundensatz	Der Teil der Daten, der in einer bestimmten Abrufanforderung (PUT oder) mit S3 Express One Zone-Speicher 512 KB überschreitet COPY
<i>region</i> -Retrieval-ZIA	GB	Stundensatz	Die Menge der Daten, die aus dem S3-One-Zone-IA Speicher abgerufen wurde
<i>region</i> -S3DSSE-In-Bytes	GB	Monatlich	Die Menge der von Amazon S3 doppelt verschlüsselten Daten
<i>region</i> -S3DSSE-Out-Bytes	GB	Monatlich	Die Menge der doppelt verschlüsselten Daten, die von Amazon S3 entschlüsselt wurden
<i>region</i> -S3G-DataTransfer-In-Bytes	GB	Stundensatz	Die Datenmenge, die in Amazon S3 übertragen wird, um Objekte aus dem S3 Glacier Flexible Retrieval- oder S3 Glacier Deep Archive-Speicher wiederherzustellen

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -S3G-DataTransfer-Out-Bytes	GB	Stundensatz	Die Menge der Daten, die von Amazon S3 zum Übergang von Objekten in den Speicher S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive übertragen werden
<i>region</i> -Select-Returned-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-Speicher zurückgegeben wurden
<i>region</i> -Select-Returned-GIR-Bytes	GB	Stundensatz	Die Datenmenge, die mit Select-Anfragen vom Speicher S3 Glacier Instant Retrieval zurückgegeben wird.
<i>region</i> -Select-Returned-INT-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Intelligent-Tiering-Speicher zurückgegeben wurden
<i>region</i> -Select-Returned-SIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-IA-Speicher zurückgegeben wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Select-Returned-ZIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-One-Zone-IA-Speicher zurückgegeben wurden
<i>region</i> -Select-Scanned-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-Speicher gescannt wurden
<i>region</i> -Select-Scanned-GIR-Bytes	GB	Stundensatz	Die Datenmenge, die mit Select-Anfragen aus dem Speicher S3 Glacier Instant Retrieval gescannt wurde.
<i>region</i> -Select-Scanned-INT-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Intelligent-Tiering-Speicher gescannt wurden
<i>region</i> -Select-Scanned-SIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-IA-Speicher gescannt wurden
<i>region</i> -Select-Scanned-ZIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-One-Zone-IA-Speicher gescannt wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -Standard-Retrieval-Bytes	GB	Stundensatz	Die Anzahl der Bytes der Daten, die mit Standardanforderungen von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive abgerufen wurden
<i>region</i> -StorageAnalytics-ObjCount	Objekte	Stündlich	Die Anzahl der eindeutigen Objekte, die in jeder Storage-Class-Analysis-Konfiguration überwacht werden.
<i>region</i> -StorageLens-ObjCount	Objekte	Täglich	Die Anzahl der eindeutigen Objekte in jedem S3 Storage Lens Dashboard , die durch erweiterte Metriken und Empfehlungen von S3 Storage Lens verfolgt werden.
<i>region</i> -StorageLensFreeTier-ObjCount	Objekte	Täglich	Die Anzahl der eindeutigen Objekte in jedem S3 Storage Lens Dashboard , die durch Nutzungsmetriken von S3 Storage Lens verfolgt werden.
StorageObjectCount	Anzahl	Täglich	Die Anzahl der Objekte in einem bestimmten Bucket
<i>region</i> -TagStorage-TagHrs	Tag-Hours	Täglich	Die Gesamtzahl der Markierungen für alle Objekte im Bucket, gemeldet nach Stunde

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -TimedStorage-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3-Standardspeicher gespeichert wurden
<i>region</i> -TimedStorage-GDA-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Deep Archive Archive-Speicher gespeichert wurden
<i>region</i> -TimedStorage-GDA-Staging	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Deep Archive Archive-Staging-Speicher gespeichert wurden
<i>region</i> -TimedStorage-GIR-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Instant Retrieval-Speicher gespeichert wurden.
<i>region</i> -TimedStorage-GIR-SmObjects	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen kleine Objekte (kleiner als 128 KB) im S3 Glacier Instant Retrieval-Speicher gespeichert wurden.
<i>region</i> -TimedStorage-GlacierByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Flexible Retrieval-Speicher gespeichert wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -TimedStorage-GlacierStaging	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Flexible Retrieval Staging-Speicher gespeichert wurden
<i>region</i> -TimedStorage-INT-FA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten auf der Stufe für häufigen Zugriff des S3 Intelligent-Tiering-Speichers 5 gespeichert wurden
<i>region</i> -TimedStorage-INT-IA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten auf der Stufe „Infrequent Access“ des S3 Intelligent-Tiering-Speichers gespeichert wurden
<i>region</i> -TimedStorage-INT-AA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten in der Archive Access-Stufe des S3 Intelligent-Tiering-Speichers gespeichert wurden
<i>region</i> -TimedStorage-INT-AIA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten in der Archive Instant Access-Stufe des S3 Intelligent-Tiering-Speichers gespeichert wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -TimedStorage-INT-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten in der Deep Archive Access-Stufe des S3 Intelligent-Tiering-Speichers gespeichert wurden
<i>region</i> -TimedStorage-RRS-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im Speicher mit reduzierter Redundanz () gespeichert wurden RRS
<i>region</i> -TimedStorage-SIA-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3 Standard-IA-Speicher gespeichert wurden
<i>region</i> -TimedStorage-SIA-SmObjects	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen kleine Objekte (kleiner als 128 KB) im S3 Standard-IA-Speicher 4 gespeichert wurden
<i>region</i> -TimedStorage-XZ-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3 Express One Zone-Speicher gespeichert wurden
<i>region</i> -TimedStorage-ZIA-ByteHrs	GB-Monat	Taglich	Die Anzahl der GB-Monate, in denen Daten im S3 One Zone-IA-Speicher gespeichert wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
<i>region</i> -TimedStorage-ZIA-SmObjects	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen kleine Objekte (kleiner als 128 KB) im S3 One Zone-IA-Speicher gespeichert wurden
<i>region</i> -Upload-XZ	GB	Stundensatz	Die Datenmenge, die in einer bestimmten Upload-Anfrage (PUT oder COPY) mit S3 Express One Zone 512 KB überschreitet

Hinweise

1. Wenn Sie eine Übertragung vor dem Abschluss beenden, kann die Menge der übertragenen Daten die Menge der von Ihrer Anwendung erhaltenen Daten überschreiten. Diese Diskrepanz kann auftreten, weil eine Anfrage zur Beendigung einer Übertragung nicht sofort ausgeführt werden kann und ein Teil der Datenmenge möglicherweise gerade übertragen wird, bis die Abbruchanfrage ausgeführt wird. Diese übertragenen Daten werden als „ausgehende“ Daten in Rechnung gestellt.
2. Wenn Objekte, die in der Speicherklasse S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert wurden, gelöscht, überschrieben oder in eine andere Speicherklasse verschoben werden, bevor die Mindestspeicherverpflichtung abgelaufen ist, die 90 Tage für S3 Glacier Instant Retrieval und S3 Glacier Flexible Retrieval oder 180 Tage für S3 Glacier Deep Archive beträgt, fällt eine anteilige Gebühr pro Gigabyte für die verbleibenden Tage an.
3. Für Objekte, die sich im S3 Standard-IA- oder S3 One Zone-IA-Speicher befinden, fällt für die verbleibenden Tage eine anteilige Gebühr pro Gigabyte an, wenn sie vor 30 Tagen gelöscht, überschrieben oder in eine andere Speicherklasse übertragen werden.
4. Für kleine Objekte (kleiner als 128 KB), die sich im S3 Standard-IA- oder S3 One Zone-IA-Speicher befinden, fällt für die verbleibenden Tage eine anteilige Gebühr pro Gigabyte an, wenn sie vor 30 Tagen gelöscht, überschrieben oder in eine andere Speicherklasse übertragen werden.

5. In der S3-Intelligent-Tiering-Speicherklasse gibt es keine mindest-abrechnungsfähige Objektgröße. Objekte, die kleiner als 128 KB sind, werden nicht überwacht oder können nicht automatisch gestuft werden. Kleinere Objekte werden immer in der S3-Intelligent-Tiering-Frequent-Access-Stufe gespeichert.
6. Wenn Sie eine `CreateMultipartUploadUploadPart`, oder `UploadPartCopy` - Anforderung für die Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive initiieren, werden Anfragen zu den S3 Standard-Anforderungstarifen abgerechnet, bis Sie den mehrteiligen Upload abgeschlossen haben. Nach Abschluss des Uploads wird für die einzelne `CompleteMultipartUpload` Anfrage der PUT Tarif für den S3 Glacier-Zielspeicher berechnet. In Bearbeitung befindliche mehrteilige Uploads für die Speicherklasse S3 Glacier Flexible Retrieval werden PUT bis zum Abschluss des Uploads als S3 Glacier Flexible Retrieval Staging Storage zu den S3-Standardspeichertarifen abgerechnet. In ähnlicher Weise werden Teile, die in Bearbeitung sind, mehrteilige Uploads für eine PUT Speicherklasse in die Speicherklasse S3 Glacier Deep Archive als S3 Glacier Deep Archive Staging Storage zu den Speichertarifen S3 Standard abgerechnet, bis der Upload abgeschlossen ist.
7. S3 Express One Zone berechnet für Anforderungsgrößen bis zu 512 KB eine Pauschalgebühr pro Anfrage. Für PUT Anfragen und GET Anfragen für den Teil der Anfrage, der größer als 512 KB ist, wird eine zusätzliche Gebühr pro GB erhoben.
8. Informationen zu den unterstützten Features der Speicherklasse S3 Express One Zone finden Sie unter [Amazon-S3-Features, die von S3 Express One Zone nicht unterstützt werden](#).
9. Nutzungstypen mit Einheiten, die in GB abgerechnet werden, werden in den Nutzungsberichten in Byte berechnet.
- 10 Ein GB-Monat wird abgeleitet, indem die Gesamtzahl der GB-Stunden über einen Monat aggregiert und dann durch die Anzahl der Stunden in diesem Monat dividiert wird. Weitere Informationen finden Sie unter [Häufig gestellte Fragen: Wie wird mir meine Nutzung von Amazon S3 in Rechnung gestellt und abgerechnet?](#)

Note

Im Allgemeinen werden den Besitzern von S3-Buckets Anfragen mit HTTP 200 OK erfolgreichen Antworten und HTTP 4XX Kundenfehlerantworten in Rechnung gestellt. Bucket-Besitzern werden keine HTTP 5XX Serverfehlerantworten, wie HTTP 503 Slow Down z. B. Fehler, in Rechnung gestellt. Weitere Informationen zu S3-Fehlercodes unter HTTP 3XX und zu 4XX Statuscodes, die nicht in Rechnung gestellt werden, finden Sie unter [Abrechnung für Amazon S3 S3-Fehlerantworten](#) Weitere Informationen zu Abrechnungsgebühren, wenn

Ihr Bucket als Bucket konfiguriert ist, für den der Antragsteller zahlt, finden Sie unter. [Die Gebühren bei Zahlung durch den Anforderer](#)

Nachverfolgen von Vorgängen in Ihren Nutzungsberichten

Operationen beschreiben die Aktion, die der angegebene Nutzungstyp an Ihrem AWS Objekt oder Bucket ausgeführt hat. Vorgänge werden durch selbsterklärende Codes beschrieben, z. B. PutObject oder ListBucket. Um zu prüfen, welche Aktionen für Ihren Bucket einen bestimmten Nutzungstyp generiert haben, verwenden Sie diese Codes. Wenn Sie einen Nutzungsbericht erstellen, können Sie wählen, All Operations aufzunehmen oder eine bestimmte Operation, z. B. GetObject, für die ein Bericht erstellt werden soll.

Weitere Informationen

- [AWS Nutzungsbericht für Amazon S3](#)
- [AWS Billing Berichte für Amazon S3](#)
- [Amazon S3 – Preise](#)
- [Amazon S3 FAQs](#)

Abrechnung für Amazon S3 S3-Fehlerantworten

Im Allgemeinen werden S3-Bucket-Besitzern Anfragen mit HTTP 200 OK erfolgreichen Antworten und HTTP 4XX Kundenfehlerantworten in Rechnung gestellt. Bucket-Besitzern werden keine HTTP 5XX Serverfehlerantworten, wie HTTP 503 Slow Down z. B. Fehler, in Rechnung gestellt. Weitere Informationen zu Abrechnungsgebühren, wenn Ihr Bucket als Bucket konfiguriert ist, für den der Antragsteller zahlt, finden Sie unter. [Die Gebühren bei Zahlung durch den Anforderer](#)

In der folgenden Tabelle sind spezifische Fehlercodes HTTP 3XX und 4XX Statuscodes aufgeführt, die nicht in Rechnung gestellt werden. Für Buckets, die mit Website-Hosting konfiguriert sind, fallen weiterhin Gebühren für Anfragen und andere Gebühren an, wenn S3 ein [benutzerdefiniertes Fehlerdokument](#) zurückgibt, oder für benutzerdefinierte Weiterleitungen.

Note

Für AccessDenied (HTTP403 Forbidden) berechnet S3 dem Bucket-Besitzer keine Gebühren, wenn die Anfrage außerhalb des individuellen AWS Kontos des Bucket-Besitzers oder der AWS Organisation des Bucket-Besitzers initiiert wird.

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes
301 Moved Permanent ly (301 Dauerhaft verschobe n)	PermanentRedirect	Der Bucket, auf den Sie zugreifen möchten, muss über den angegebenen Endpunkt adressiert werden. Senden Sie alle future Anfragen an diesen Endpunkt.
	PermanentRedirectC ontrolError	Der API Vorgang, auf den Sie zugreifen möchten, muss über den angegebenen Endpunkt ausgeführt werden. Senden Sie alle future Anfragen an diesen Endpunkt.
307 Temporäre Weiterlei tung	TemporaryRedirect	Sie werden zum Bucket umgeleitet, während der Domain Name System (DNS) -Server aktualisiert wird.
400 Ungültige Anfrage	AuthorizationHeade rMalformed	Der von Ihnen angegebene Autorisierungsheader ist ungültig.

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes
	AuthorizationQuery ParametersError	Die von Ihnen angegebenen Autorisierungsabfrageparameter sind nicht gültig.
	ConnectionClosedBy Requester	Wird an den ursprünglichen Aufrufer zurückgegeben, wenn beim Lesen des WriteGetObjectResponse Hauptteils ein Fehler aufgetreten ist.
	DeviceNotActiveError	Das Gerät ist derzeit nicht aktiv.
	EndpointNotFound	Direkte Anfragen an den richtigen Endpunkt.
	ExpiredToken	Das bereitgestellte Token ist abgelaufen.
	IllegalLocationConstraintException	Sie versuchen, auf einen Bucket aus einer anderen Region als der Region zuzugreifen, in der der Bucket existiert. Verwenden Sie die <code>--region</code> Option, um diesen Fehler zu vermeiden. Beispiel: <pre>aws s3 cp <i>awsexample.txt</i> s3://<i>amzn-s3-demo-bucket</i> / --region <i>ap-east-1</i> .</pre>

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidArgument	<p>Dieser Fehler kann aus folgenden Gründen auftreten:</p> <ul style="list-style-type: none">• Das angegebene Argument war nicht gültig.• In der Anfrage fehlte ein erforderlicher Header.• Das angegebene Argument war unvollständig oder hatte das falsche Format.• Das angegebene Argument muss eine Länge größer oder gleich 3 haben.	
	InvalidBucketOwner AWSAccountID	Der Wert des Parameters Expected Bucket Owner muss eine AWS-Konto ID sein.	
	InvalidDigest	Der von Ihnen angegebene Inhalts- MD5 oder Prüfsummenwert ist nicht gültig.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidEncryptionA lgorithmError	Die von Ihnen angegeben e Verschlüsselungsan frage ist nicht gültig. Der gültige Wert lautet AES256.	
	InvalidHostHeader	Die in der Anfrage angegebenen Host- Header verwendeten die falsche Adressierung.	
	InvalidHttpMethod	Die Anfrage wird mit einer unerwarteten HTTP Methode gestellt.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidRequest	<p>Dieser Fehler kann aus folgenden Gründen auftreten:</p> <ul style="list-style-type: none">• Die Anfrage verwendet die falsche Signaturversion. Verwenden Sie AWS4-HMAC-SHA256 (Signaturversion 4).• Ein Access Point kann nur für einen vorhandenen Bucket erstellt werden.• Der Access Point befindet sich nicht in einem Zustand, in dem er gelöscht werden kann.• Ein Access Point kann nur für einen vorhandenen Bucket aufgeführt werden.• Das nächste Token ist nicht gültig.• In einer Lebenszyklusregel muss mindestens eine Aktion angegeben werden.•	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
		<p>Es muss mindestens eine Lebenszyklusregel angegeben werden.</p> <ul style="list-style-type: none">• Die Anzahl der Lebenszyklusregeln darf die zulässige Grenze von 1000 Regeln nicht überschreiten.• Der Bereich für den MaxResults Parameter ist nicht gültig.• SOAPAnfragen müssen über eine HTTPS Verbindung gestellt werden.• Amazon S3 Transfer Acceleration wird für Buckets mit nicht DNS konformen Namen nicht unterstützt.• Amazon S3 Transfer Acceleration wird für Buckets mit Punkten (.) im Namen nicht unterstützt.•	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
		<p>Der Amazon S3 Transfer Acceleration Acceleration-Endpunkt unterstützt nur Anfragen im virtuellen Stil.</p> <ul style="list-style-type: none">• Amazon S3 Transfer Acceleration ist für diesen Bucket nicht konfiguriert.• Amazon S3 Transfer Acceleration ist in diesem Bucket deaktiviert.• Amazon S3 Transfer Acceleration wird in diesem Bucket nicht unterstützt. Wenn Sie Hilfe benötigen, wenden Sie sich an AWS Support.• Amazon S3 Transfer Acceleration kann für diesen Bucket nicht aktiviert werden. Wenn Sie Hilfe benötigen, wenden Sie sich an AWS Support.•	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
		<p>Widersprüchliche Werte in HTTP Headern und Abfrageparametern.</p> <ul style="list-style-type: none">• Widersprüchliche Werte in HTTP Kopfzeilen und Formularfeldern. POST• CopyObject Anfrage für Objekte mit einer Größe von mehr als 5 GB.	
	InvalidSessionException	Wird zurückgegeben, wenn die Sitzung nicht mehr existiert, weil das Zeitlimit überschritten wurde oder sie abgelaufen ist.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidSignature	Die vom Server berechnete Anforderungssignatur stimmt nicht mit der von Ihnen angegebenen Signatur überein. Überprüfen Sie Ihren AWS geheimen Zugriffsschlüssel und Ihre Signaturmethode. Weitere Informationen finden Sie unter RESTAnfragen signieren und authentifizieren .	
	InvalidSOAPRequest	Der SOAP Text der Anfrage ist nicht gültig.	
	InvalidStorageClass	Die von Ihnen angegebene Speicherklasse ist ungültig.	
	InvalidTag	Ihre Anfrage enthält eine ungültige Tag-Eingabe. Ihre Anfrage könnte beispielsweise doppelte Schlüssel, Schlüssel oder Werte, die zu lang sind, oder Systemtags enthalten.	
	InvalidToken	Das angegebene Token ist falsch formatiert oder anderweitig nicht gültig.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	Ungültig URI	Die angegebene URI Datei konnte nicht analysiert werden.	
	KeyTooLongError	Ihr Schlüssel ist zu lang.	
	KMS.DisabledException	Die Anfrage wurde abgelehnt, weil der angegebene KMS Schlüssel nicht aktiviert ist.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	KMS.InvalidKeyUsag eException	<p>Die Anfrage wurde aus einem der folgenden Gründe abgelehnt:</p> <ul style="list-style-type: none"> • Der KeyUsage Wert des KMS Schlüssels ist mit dem API Vorgang nicht kompatibel. • Der für den Vorgang angegebene Verschlüsselungs- oder Signaturalgorithmus ist nicht mit dem Typ des Schlüsselmaterials im KMS Schlüssel (KeySpec) kompatibel. <p>Zum Verschlüsseln, Entschlüsseln, erneuten Verschlüsseln und Generieren von Datenschlüsseln muss der KeyUsage Wert <code>_</code> sein. ENCRYPT DECRYPT Zum Signieren und Überprüfen von Nachrichten muss der Wert <code>_</code> sein. KeyUsage SIGN VERIFY Zum Generieren und Überprüfen von Nachrichtenauthentifizierungscodes</p>	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
		<p>(MACs) KeyUsage muss GENERATE _ _ VERIFY sein. MAC Um wichtige Vertragsg heimnisse abzuleiten, KeyUsage muss der Wert _ seinKEY. AGREEMENT Verwenden Sie die DescribeKey Operation, um den Wert KeyUsage eines KMS Schlüssels zu ermitteln. Verwenden Sie die DescribeKey Operation, um die Verschlüsselungs- oder Signierungsalgorit hmen zu finden, die für einen bestimmten KMS Schlüssel unterstützt werden.</p>	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	KMS.KMSInvalidStateException	<p>Die Anfrage wurde abgelehnt, da der Status der angegebenen Ressource für diese Anfrage nicht gültig ist. Diese Ausnahme bedeutet eine der folgenden Ursachen:</p> <ul style="list-style-type: none">• Der Schlüsselstatus des KMS Schlüssels ist nicht mit dem Vorgang kompatibel. <p>Verwenden Sie die DescribeKey Operation , um den Schlüssel status zu ermitteln . Weitere Informationen darüber, welche Schlüsselstatus mit den einzelnen KMS Vorgängen kompatibel sind, finden Sie AWS KMS keys im AWS Key Management Service Entwicklerhandbuch unter Schlüsselstatus von.</p> <ul style="list-style-type: none">• Bei kryptografischen Vorgängen mit KMS Schlüsseln in benutzerdefinierten Schlüsselspeichern	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
		stellt diese Ausnahme einen allgemeinen Fehler mit vielen möglichen Ursachen dar. Informationen zur Identifizierung der Ursache finden Sie in der Fehlermeldung, die der Ausnahme beiliegt.	
	KMS.NotFoundException	Die Anfrage wurde abgelehnt, weil die angegebene Entität oder Ressource nicht gefunden werden konnte.	
	LambdaInvalidResponse	Kehrt zum ursprünglichen Anrufer zurück, wenn er mit WriteGetObjectResponse ValidationError auf AWS Lambda antwortet . Weitere Informationen finden Sie in der ValidationError Nachricht . Nicht alle Fälle ValidationError führen zu einem LambdaInvalidResponse Fehler.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	LambdaInvocationFailed	Der Aufruf der Lambda-Funktion ist fehlgeschlagen. Aufrufer erhalten möglicherweise den folgenden Fehler, wenn S3 Object Lambda die konfigurierte Lambda-Funktion nicht erfolgreich aufrufen kann. Die Fehlermeldung kann Details zu einem eventuellen Fehler enthalten, den der AWS Lambda Dienst beim Aufrufen der Funktion zurückgibt (z. B. Statuscode, Fehlercode, Fehlermeldung und Anforderungs-ID).	
	M alformedACLError	Die von ACL Ihnen angegebene Datei war falsch formatiert oder wurde nicht anhand unseres veröffentlichten Schemas validiert.	
	M alformedPOSTRequest	Der Hauptteil Ihrer POST Anfrage besteht nicht aus wohlgeformten Multipart/Form-Daten.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes
	Falsch formatiert XML	Die von XML Ihnen angegebene Datei war falsch formatiert oder wurde nicht anhand unseres veröffentlichten Schemas validiert.
	MaxPostPreDataLengthExceededError	Ihre POST Anforderungsfelder vor der Upload-Datei waren zu groß.
	MetadataTooLarge	Ihre Metadaten-Header überschreiten die maximal zulässige Metadatengröße.
	MissingAttachment	Es wurde ein SOAP Anhang erwartet, aber keiner gefunden.
	MissingRequestBodyError	Sie haben ein leeres XML Dokument als Anfrage gesendet.
	MissingSecurityHeader	In Ihrer Anfrage fehlt ein erforderlicher Header.
	NoLoggingStatusForKey	So etwas wie eine Unterressource für den Protokollierungsstatus eines Schlüssels gibt es nicht.

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	NotDeviceOwnerError	Das Gerät, das das Token generiert hat, gehört nicht dem authentifizierten Benutzer.	
	ResponseInterrupted	Wird an den ursprünglichen Anrufer zurückgesendet, wenn beim Lesen des WriteGetObjectResponse Textes ein Fehler auftritt.	
	RequestHeaderSectionTooLarge	Der Anforderungsheader und die Abfrageparameter, mit denen die Anfrage gestellt wurde, überschreiten die maximal zulässigen Größen	
	TokenCodeInvalidError	Die von Ihnen angegebene Seriennummer und/oder der Token-Code sind nicht gültig.	
	UnexpectedContent	Diese Anfrage enthält Inhalte, die nicht unterstützt werden.	
	UnsupportedArgument	Die Anfrage enthielt ein nicht unterstütztes Argument.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	UnsupportedSignature	Die bereitgestellte Anfrage ist mit einer nicht unterstützten STS Token-Version signiert, oder die Signaturversion wird nicht unterstützt.	
	UserKeyMustBeSpecified	Die POST Bucket-Anfrage muss den angegebenen Feldnamen enthalten. Wenn es angegeben ist, überprüfen Sie die Reihenfolge der Felder.	
	IncorrectEndpoint	Der angegebene Bucket ist in einer anderen Region vorhanden. Direkte Anfragen an den richtigen Endpunkt.	
	ValidationError	Bei der WriteGetObjectResponse API Operation können Validierungsfehler auftreten, die aus zahlreichen Gründen auftreten können. Weitere Informationen finden Sie in der Fehlermeldung.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
403 Verboten	RequestTimeTooSkewed	Der Unterschied zwischen der Anforderungszeit und der Serverzeit ist zu groß.	
	SignatureDoesNotMatch	Die vom Server berechnete Anforderungssignatur stimmt nicht mit der von Ihnen angegebenen Signatur überein. Überprüfen Sie Ihren AWS geheimen Zugriffsschlüssel und Ihre Signaturmethode. Weitere Informationen finden Sie unter RESTAuthentifizierung und SOAPAuthentifizierung .	
	NotSignedUp	Ihr Konto ist nicht für den Amazon S3 S3-Service registriert. Sie müssen sich registrieren, bevor Sie Amazon S3 verwenden können. Sie können sich wie folgt registrierenURL: https://aws.amazon.com/s3	
	InvalidSecurity	Die angegebenen Sicherheitsanmeldedaten sind nicht gültig.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidPayer	Der gesamte Zugriff auf dieses Objekt wurde deaktiviert. Weitere Unterstützung finden Sie unter Kontaktieren Sie uns .	
	InvalidAccessKeyId	Die von Ihnen angegebene AWS Zugriffsschlüssel-ID ist in unseren Aufzeichnungen nicht vorhanden.	
	AccountProblem	Es liegt ein Problem mit Ihrem vor AWS-Konto, das verhindert, dass der Vorgang erfolgreich abgeschlossen werden kann. Weitere Unterstützung finden Sie unter Kontaktieren Sie uns .	
	UnauthorizedAccessError	Gilt nur in chinesischen Regionen. Wird zurückgegeben, wenn eine Anfrage an einen Bucket gestellt wird, der keine ICP Lizenz hat. Weitere Informationen finden Sie unter ICPRecordal .	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	UnexpectedIPError	Gilt nur in chinesischen Regionen. Diese Anfrage wurde abgelehnt, da die IP unerwartet war.	
	MissingAuthenticationToken	Die Anfrage wurde nicht signiert.	
	LambdaPermissionError	Der Aufrufer ist nicht berechtigt, die Lambda-Funktion aufzurufen. Der Aufrufer muss die Erlaubnis haben, die Lambda-Funktion aufzurufen. Überprüfen Sie die dem Aufrufer beigefügten Richtlinien und stellen Sie sicher, dass sie <code>lambda:Invoke</code> für die konfigurierte Funktion verwendet werden dürfen. Die Fehlermeldung kann Details zu einem eventuellen Fehler enthalten, den der Lambda-Dienst beim Aufrufen der Funktion zurückgibt (z. B. Statuscode, Fehlercode, Fehlermeldung und Anforderungs-ID).	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
404 Not Found (404 Nicht gefunden)	LambdaNotFound	Die AWS Lambda Funktion wurde nicht gefunden. Die konfigurierte Lambda-Funktion, Version oder Alias wurde beim Versuch, sie aufzurufen, nicht gefunden. Stellen Sie sicher, dass die Konfiguration des S3 Object Lambda Access Points auf die richtige Lambda-Funktion verweist. ARN Die Fehlermeldung kann Details zu einem eventuell en Fehler enthalten, den der AWS Lambda Dienst beim Aufrufen der Funktion zurückgibt (z. B. Statuscode, Fehlercode, Fehlermeldung und Anforderungs-ID).	
	NoSuchAsyncRequest	Die angegebene Anfrage wurde nicht gefunden.	
	NoSuchObjectLockConfiguration	Das angegebene Objekt hat keine ObjectLock Konfiguration.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes	
	NoSuchUpload	Der angegebene mehrteilige Upload ist nicht vorhanden. Die Upload-ID ist möglicherweise nicht gültig, oder der mehrteilige Upload wurde möglicherweise abgebrochen oder abgeschlossen.	
	NoSuchWebsiteConfiguration	Der angegebene Bucket hat keine Website-Konfiguration.	
	NoTransformationDefined	Für diesen Object Lambda Access Point wurde keine Transformation gefunden.	
	ObjectLockConfigurationNotFoundError	Die Object Lock-Konfiguration ist für diesen Bucket nicht vorhanden.	
405 Methode nicht zulässig	MethodNotAllowed	Die angegebene Methode ist für diese Ressource nicht zulässig.	

HTTPStatu scode	Fehlercode	Beschreibung des Fehlercodes
409 Konflikt	BucketAlreadyExists	Der angeforderte Bucket-Name ist nicht verfügbar . Der Bucket-Namespacer wird von allen Benutzern des Systems gemeinsam genutzt. Geben Sie einen anderen Namen an und versuchen Sie es erneut.
	InvalidBucketState	Die Anfrage ist für den aktuellen Status des Buckets nicht gültig.
	OperationAborted	Für diese Ressource wird derzeit ein bedingter Vorgang ausgeführt, bei dem ein Konflikt aufgetreten ist. Versuchen Sie es erneut.
411 Länge erforderlich	MissingContentLength	Sie müssen den HTTP Content-Length-Header angeben.
4.1.2 Die Vorbedingung ist fehlgeschlagen	RequestIsNotMultipartContent	Bei einer POST Bucket-Anfrage muss es sich um eine Anlage vom Typ Multipart/Form-Data handeln.
4.1.6 Der angeforderte Bereich ist nicht erfüllbar	InvalidRange	Der angeforderte Bereich ist für die Anfrage nicht gültig. Versuchen Sie es mit einem anderen Bereich.

Filtern und Abrufen von Daten mit Amazon S3 Select

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Mit Amazon S3 Select können Sie Structured Query Language (SQL) -Anweisungen verwenden, um den Inhalt eines Amazon S3 S3-Objekts zu filtern und nur die Teilmenge der Daten abzurufen, die Sie benötigen. Wenn Sie die Daten mit Amazon S3 Select filtern, wird die von Amazon S3 übertragene Datenmenge reduziert, wodurch sich die Kosten und Latenzzeiten für diesen Datenabruf verringern.

Mit Amazon S3 Select können Sie jeweils nur ein Objekt abfragen. Es funktioniert mit einem Objekt, das im CSVJSON, oder Apache Parquet Format gespeichert ist. Es funktioniert auch mit Objekten, die mit GZIP oder komprimiert sind BZIP2 (nur für JSON Objekte CSV und), und mit serverseitig verschlüsselten Objekten. Sie können das Format der Ergebnisse entweder als CSV oder angeben und festlegenJSON, wie die Datensätze im Ergebnis getrennt werden.

In der Anfrage übergeben Sie SQL Ausdrücke an Amazon S3. Amazon S3 Select unterstützt eine Teilmenge von. SQL Weitere Informationen zu den SQL Elementen, die von Amazon S3 Select unterstützt werden, finden Sie unter [SQLReferenz für Amazon S3 Select](#).

Sie können SQL Abfragen mithilfe der Amazon S3 S3-Konsole, der AWS Command Line Interface (AWS CLI), der `SelectObjectContent` REST API Operation oder der ausführen AWS SDKs.

Note

Bei der Amazon S3-Konsole ist die zurückgegebene Datenmenge auf 40 MB begrenzt. Um weitere Daten abzurufen, verwenden Sie das AWS CLI oder dasAPI.

Voraussetzungen und Einschränkungen

Für die Verwendung von Amazon S3 Select müssen folgende Voraussetzungen erfüllt sein:

- Sie haben die Berechtigung `s3:GetObject` für das abzufragende Objekt.

- Wenn das Objekt, das Sie abfragen, mit serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verschlüsselt ist, müssen Sie den Verschlüsselungsschlüssel verwenden `https` und in der Anfrage angeben.

Für die Verwendung von Amazon S3 Select gelten die folgenden Einschränkungen:

- S3 Select kann nur ein Objekt pro Anfrage abfragen.
- Die maximale Länge eines SQL Ausdrucks beträgt 256 KB.
- Die Maximallänge eines Datensatzes in der Eingabe oder im Ergebnis liegt bei 1 MB.
- Amazon S3 Select kann nur verschachtelte Daten mithilfe des JSON Ausgabeformats ausgeben.
- Sie können kein Objekt abfragen, das in den Speicherklassen S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive oder Reduced Redundancy Storage (RRS) gespeichert ist. Sie können auch kein Objekt abfragen, das auf der Stufe S3 Intelligent-Tiering Archive Access oder der Stufe S3 Intelligent-Tiering Deep Archive Access gespeichert ist. Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Bei der Verwendung von Amazon S3 Select mit einem Parquet Objekt gelten zusätzliche Einschränkungen:

- Amazon S3 Select unterstützt nur die spaltenförmige Komprimierung mit GZIP oder Snappy. Amazon S3 Select unterstützt keine Komprimierung des gesamten Objekts für ein Parquet Objekt.
- Amazon S3 Select unterstützt keine Parquet-Ausgabe. Sie müssen das Ausgabeformat als CSV oder angeben. JSON
- Die Maximalgröße für unkomprimierte Zeilengruppen beträgt 512 MB.
- Sie müssen die im Schema des Objekts angegebenen Datentypen verwenden.
- Select-Anfragen für ein wiederholtes Feld geben ausschließlich den letzten Wert zurück.

Erstellen einer Anforderung

Beim Erstellen einer Anforderung geben Sie die Details des abzufragenden Objekts mithilfe eines `InputSerialization`-Objekts an. Die Details zur Rückgabe der Ergebnisse geben Sie mit einem `OutputSerialization`-Objekt an. Sie geben auch den SQL Ausdruck an, den Amazon S3 verwendet, um die Anfrage zu filtern.

Weitere Informationen zur Erstellung einer Amazon S3 Select-Anfrage finden Sie [SelectObjectContent](#) in der Amazon Simple Storage Service API Reference. In den folgenden Abschnitten finden Sie auch eines der SDK Codebeispiele.

Anforderungen mittels Scanbereichen

Mit Amazon S3 Select können Sie eine Teilmenge eines Objekts scannen, indem Sie einen abzufragenden Byte-Bereich angeben. Mit dieser Fähigkeit können Sie das gesamte Objekt parallel scannen, indem Sie die Arbeit in separate Amazon S3 Select-Anforderungen für eine Reihe von sich nicht überlappenden Scanbereichen aufteilen.

Scanbereiche müssen nicht an Datensatzgrenzen ausgerichtet werden. Eine Scanbereichsanforderung von Amazon S3 Select wird über den angegebenen Byte-Bereich ausgeführt. Ein Datensatz, der im angegebenen Scanbereich beginnt, aber darüber hinaus reicht, wird von der Abfrage bearbeitet. Im Folgenden wird beispielsweise ein Amazon S3 S3-Objekt gezeigt, das eine Reihe von Datensätzen in einem zeilengetrennten Format CSV enthält:

```
A, B  
C, D  
D, E  
E, F  
G, H  
I, J
```

Gehen wir einmal davon aus, dass Sie den Amazon-S3-Select-Parameter `ScanRange` verwenden, bei (Byte) 1 beginnen und bei (Byte) 4 enden. Der Scanbereich würde also bei „,“ beginnen und der Scan würde bis zum Ende des Datensatzes fortgesetzt, der bei „C“ beginnt. Ihre Scanbereichsanfrage gibt das Ergebnis „C, D“ zurück, da dies das Ende des Datensatzes ist.

Der Amazon S3 Select-Scanbereich fordert Unterstützung Parquet CSV (ohne Trennzeichen in Anführungszeichen) oder JSON Objekte (nur im LINES Modus) an. CSV und JSON Objekte müssen unkomprimiert sein. Wenn für zeilenbasierte JSON Objekte CSV und Objekte ein Scanbereich als Teil der Amazon S3 Select-Anfrage angegeben wird, werden alle Datensätze verarbeitet, die innerhalb des Scanbereichs beginnen. Bei Parquet-Objekten werden alle Zeilengruppen, die innerhalb des angeforderten Scanbereichs beginnen, verarbeitet.

Amazon S3 Select-Scanbereichsanforderungen können mit Amazon S3 API und verwendet AWS SDKs werden. AWS CLI Sie können den `ScanRange`-Parameter in der Amazon S3 Select-Anfrage als diese Funktion verwenden. Weitere Informationen finden Sie [SelectObjectContent](#) in der Amazon Simple Storage Service API Reference.

Fehler

Amazon S3 Select gibt einen Fehlercode und eine zugehörige Fehlermeldung zurück, wenn beim Versuch, eine Abfrage auszuführen, ein Problem auftritt. Eine Liste der Fehlercodes und Beschreibungen finden Sie im Abschnitt [Liste der Fehlercodes für SELECT Objektinhalte](#) auf der Seite Error Responses in der Amazon Simple Storage Service API Reference.

Weitere Informationen zu Amazon S3 Select finden Sie in den folgenden Themen:

Themen

- [Beispiele für die Verwendung von Amazon S3 Select für ein Objekt](#)
- [SQLReferenz für Amazon S3 Select](#)

Beispiele für die Verwendung von Amazon S3 Select für ein Objekt

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Sie können S3 Select verwenden, um Inhalte aus einem Objekt auszuwählen, indem Sie die Amazon S3 S3-Konsole RESTAPI, die und die verwenden AWS SDKs.

Weitere Informationen zu den unterstützten SQL Funktionen für S3 Select finden Sie unter [SQLFunktionen](#).

Verwenden der S3-Konsole

Inhalte aus einem Objekt in der Amazon S3-Konsole auswählen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie den Bucket aus, der das Objekt enthält, aus dem Sie Inhalte auswählen möchten, und wählen Sie dann den Namen des Objekts aus.
4. Wählen Sie Objektaktionen und anschließend Abfragen mit S3 Select aus.
5. Konfigurieren Sie die Eingabeeinstellungen basierend auf dem Format Ihrer Eingabedaten.

6. Konfigurieren Sie die Ausgabeeinstellungen basierend auf dem Format der Ausgabe, die Sie empfangen möchten.
7. Um Datensätze aus dem ausgewählten Objekt zu extrahieren, geben Sie unter SQLAbfrage die SELECT SQL Befehle ein. Weitere Informationen zum Schreiben von SQL Befehlen finden Sie unter [SQLReferenz für Amazon S3 Select](#).
8. Nachdem Sie SQL Abfragen eingegeben haben, wählen Sie SQLAbfrage ausführen. Anschließend können Sie unter Abfrageergebnisse die Ergebnisse Ihrer SQL Abfragen sehen.

Verwenden Sie den REST API

Sie können den verwenden AWS SDKs, um Inhalt aus einem Objekt auszuwählen. Wenn Ihre Anwendung dies jedoch erfordert, können Sie REST Anfragen direkt senden. Weitere Informationen über das Anforderungs- und Antwort-Format finden sie unter [SelectObjectContent](#).

Mit dem AWS SDKs

Sie können Amazon S3 Select verwenden, um mithilfe der `selectObjectContent` Methode einen Teil des Inhalts eines Objekts auszuwählen. Wenn diese Methode erfolgreich ist, gibt sie die Ergebnisse des SQL Ausdrucks zurück.

Java

Der folgende Java-Code gibt den Wert der ersten Spalte für jeden Datensatz zurück, der in einem Objekt gespeichert ist, das Daten enthält, die im CSV Format gespeichert sind. Im Beispiel müssen auch Progress- und Stats-Meldungen zurückgegeben werden. Sie müssen einen gültigen Bucket-Namen und ein Objekt angeben, das Daten im CSV Format enthält.

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
package com.amazonaws;

import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CSVInput;
import com.amazonaws.services.s3.model.CSVOutput;
import com.amazonaws.services.s3.model.CompressionType;
import com.amazonaws.services.s3.model.ExpressionType;
import com.amazonaws.services.s3.model.InputSerialization;
import com.amazonaws.services.s3.model.OutputSerialization;
```

```
import com.amazonaws.services.s3.model.SelectObjectContentEvent;
import com.amazonaws.services.s3.model.SelectObjectContentEventVisitor;
import com.amazonaws.services.s3.model.SelectObjectContentRequest;
import com.amazonaws.services.s3.model.SelectObjectContentResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.concurrent.atomic.AtomicBoolean;

import static com.amazonaws.util.IOUtils.copy;

/**
 * This example shows how to query data from S3Select and consume the response in
 * the form of an
 * InputStream of records and write it to a file.
 */

public class RecordInputStreamExample {

    private static final String BUCKET_NAME = "${my-s3-bucket}";
    private static final String CSV_OBJECT_KEY = "${my-csv-object-key}";
    private static final String S3_SELECT_RESULTS_PATH = "${my-s3-select-results-
path}";
    private static final String QUERY = "select s._1 from S3Object s";

    public static void main(String[] args) throws Exception {
        final AmazonS3 s3Client = AmazonS3ClientBuilder.defaultClient();

        SelectObjectContentRequest request = generateBaseCSVRequest(BUCKET_NAME,
CSV_OBJECT_KEY, QUERY);
        final AtomicBoolean isResultComplete = new AtomicBoolean(false);

        try (OutputStream fileOutputStream = new FileOutputStream(new File
(S3_SELECT_RESULTS_PATH));
            SelectObjectContentResult result =
s3Client.selectObjectContent(request)) {
            InputStream resultInputStream =
result.getPayload().getRecordsInputStream(
                new SelectObjectContentEventVisitor() {
                    @Override
                    public void visit(SelectObjectContentEvent.StatsEvent event)
                    {

```

```

        System.out.println(
            "Received Stats, Bytes Scanned: " +
event.getDetails().getBytesScanned()
            + " Bytes Processed: " +
event.getDetails().getBytesProcessed());
    }

    /*
     * An End Event informs that the request has finished
successfully.
     */
    @Override
    public void visit(SelectObjectContentEvent.EndEvent event)
    {
        isResultComplete.set(true);
        System.out.println("Received End Event. Result is
complete.");
    }
    }

    );

    copy(resultInputStream, fileOutputStream);
}

/*
 * The End Event indicates all matching records have been transmitted.
 * If the End Event is not received, the results may be incomplete.
 */
if (!isResultComplete.get()) {
    throw new Exception("S3 Select request was incomplete as End Event was
not received.");
}
}

private static SelectObjectContentRequest generateBaseCSVRequest(String bucket,
String key, String query) {
    SelectObjectContentRequest request = new SelectObjectContentRequest();
    request.setBucketName(bucket);
    request.setKey(key);
    request.setExpression(query);
    request.setExpressionType(ExpressionType.SQL);

    InputSerialization inputSerialization = new InputSerialization();
    inputSerialization.setCsv(new CSVInput());
}

```

```
    inputSerialization.setCompressionType(CompressionType.NONE);
    request.setInputSerialization(inputSerialization);

    OutputSerialization outputSerialization = new OutputSerialization();
    outputSerialization.setCsv(new CSVOutput());
    request.setOutputSerialization(outputSerialization);

    return request;
}
}
```

JavaScript

Ein JavaScript Beispiel, bei dem der `SelectObjectContent` API Vorgang AWS SDK for JavaScript with the S3 verwendet wird, um Datensätze JSON und CSV Dateien auszuwählen, die in Amazon S3 gespeichert sind, finden Sie im Blogbeitrag [Einführung in die Unterstützung von Amazon S3 Select in der AWS SDK for JavaScript](#).

Python

Ein Python-Beispiel für die Verwendung von SQL Abfragen zum Durchsuchen von Daten, die mithilfe von S3 Select als Datei mit kommagetrennten Werten (CSV) in Amazon S3 geladen wurden, finden Sie im Blogbeitrag [Abfragen von Daten ohne Server oder Datenbanken mit Amazon S3 Select](#).

SQLReferenz für Amazon S3 Select

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Diese Referenz enthält eine Beschreibung der Elemente der strukturierten Abfragesprache (SQL), die von Amazon S3 Select unterstützt werden.

Themen

- [SELECT command](#)
- [Datentypen](#)

- [Operatoren](#)
- [Reservierte Schlüsselwörter](#)
- [SQLFunktionen](#)

SELECT command

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt nur den SELECT SQL Befehl. Die folgenden ANSI Standardklauseln werden unterstützt fürSELECT:

- SELECT table
- FROM-Klausel
- WHERE-Klausel
- LIMIT-Klausel

Note

Amazon S3 Select-Abfragen unterstützen derzeit keine Unterabfragen oder Joins.

SELECT table

Die SELECT-Liste nennt die Spalten, Funktionen und Ausdrücke, die die Abfrage zurückgeben soll. Der Liste stellt die Ausgabe der Abfrage dar.

```
SELECT *  
SELECT projection1 AS column_alias_1, projection2 AS column_alias_2
```

Bei der ersten SELECT-Form mit dem * (Sternchen) werden alle Zeilen zurückgegeben, die die Bedingung der WHERE-Klausel erfüllen. Bei der zweiten SELECT-Form wird für jede Spalte eine Zeile mit benutzerdefinierten skalaren Ausgabeausdrücken *projection1* und *projection2* erstellt.

FROM-Klausel

Amazon S3 Select unterstützt die folgenden Formen der FROM-Klausel:

```
FROM table_name
FROM table_name alias
FROM table_name AS alias
```

In jeder Form der FROM-Klausel steht `table_name` für das S3Object, das abgefragt wird. Benutzer traditioneller relationaler Datenbanken können sich dies als Datenbankschema mit mehreren Ansichten einer Tabelle vorstellen.

Gemäß dem Standard SQL erstellt die FROM Klausel Zeilen, die in der WHERE Klausel gefiltert und in die SELECT Liste projiziert werden.

Für JSON Objekte, die in Amazon S3 Select gespeichert sind, können Sie auch die folgenden Formen der FROM Klausel verwenden:

```
FROM S3object[*].path
FROM S3object[*].path alias
FROM S3object[*].path AS alias
```

Mit dieser Form der FROM Klausel können Sie aus Arrays oder Objekten innerhalb eines JSON Objekts auswählen. Sie können „path“ unter Verwendung einer der folgenden Formen angeben:

- Nach Name (in einem Objekt): `.name` oder `['name']`
- Nach Index (in einem Array): `[index]`
- Nach Platzhalterzeichen (in einem Objekt): `. *`
- Nach Platzhalterzeichen (in einem Array): `[*]`

Note

- Diese Form der FROM Klausel funktioniert nur mit JSON Objekten.
- Platzhalterzeichen übermitteln stets mindestens einen Datensatz. Wenn es keinen übereinstimmenden Datensatz gibt, übermittelt Amazon S3 Select den Wert MISSING. Während der Ausgabeserialisierung (nach Abschluss der Abfrage) ersetzt Amazon S3 Select MISSING-Werte durch leere Datensätze.
- Aggregierte Funktionen (AVG, COUNT, MAX, MIN und SUM) überspringen MISSING-Werte.

- Wenn Sie bei Verwendung eines Platzhalterzeichens keinen Alias angeben, können Sie auf die Zeile verweisen, die das letzte Element im Pfad verwendet. Sie könnten beispielsweise alle Preise aus einer Liste von Büchern unter Verwendung der Abfrage `SELECT price FROM S3Object[*].books[*].price` auswählen. Wenn der Pfad mit einem Platzhalterzeichen anstelle eines Namens endet, können Sie den Wert `_1` verwenden, um auf die Zeile zu verweisen. Beispielsweise könnten Sie anstelle von `SELECT price FROM S3Object[*].books[*].price` die Abfrage `SELECT _1.price FROM S3Object[*].books[*]` verwenden.
- Amazon S3 Select behandelt ein JSON Dokument immer als ein Array von Werten auf Stammebene. Selbst wenn das JSON Objekt, das Sie abfragen, nur ein Stammelement hat, muss die FROM Klausel also mit `beginnen`. `S3Object[*]` Aus Kompatibilitätsgründen ermöglicht Amazon S3 Select Ihnen das Auslassen des Platzhalterzeichens, wenn Sie keinen Pfad einfügen. Daher ist die vollständige Klausel `FROM S3Object` gleichwertig mit `FROM S3Object[*] as S3Object`. Wenn Sie einen Pfad einfügen, müssen Sie auch das Platzhalterzeichen verwenden. Daher sind sowohl `FROM S3Object` als auch `FROM S3Object[*].path` gültige Klauseln, nicht aber `FROM S3Object.path`.

Example

Beispiele:

Beispiel 1

Dieses Beispiel zeigt Ergebnisse unter Verwendung des folgenden Datensatzes und der folgenden Abfrage:

```
{ "Rules": [ {"id": "1"}, {"expr": "y > x"}, {"id": "2", "expr": "z = DEBUG"} ] }
{ "created": "June 27", "modified": "July 6" }
```

```
SELECT id FROM S3Object[*].Rules[*].id
```

```
{"id":"1"}
{}
{"id":"2"}
{}
```

Amazon S3 Select produziert die einzelnen Ergebnisse aus den folgenden Gründen:

- {"id":"id-1"} – S3object[0].Rules[0].id produzierte eine Übereinstimmung.
- {} – S3object[0].Rules[1].id produzierte keine Übereinstimmung mit einem Datensatz. Daher übermittelte Amazon S3 Select den Wert „MISSING“, der anschließend während der Ausgabeserialisierung in einen leeren Datensatz geändert und zurückgegeben wurde.
- {"id":"id-2"} – S3object[0].Rules[2].id produzierte eine Übereinstimmung.
- {} – S3object[1] produzierte keine Übereinstimmung mit Rules. Daher übermittelte Amazon S3 Select den Wert „MISSING“, der anschließend während der Ausgabeserialisierung in einen leeren Datensatz geändert und zurückgegeben wurde.

Wenn Sie nicht möchten, dass Amazon S3 Select leere Datensätze zurückgibt, wenn keine Übereinstimmung gefunden wird, können Sie einen Test auf den -Wert ausführe MISSING. Die folgende Abfrage gibt dieselben Ergebnisse wie die vorherige Abfrage zurück, jedoch mit Auslassung der leeren Werte:

```
SELECT id FROM S3object[*].Rules[*].id WHERE id IS NOT MISSING
```

```
{"id":"1"}
{"id":"2"}
```

Beispiel 2

Dieses Beispiel zeigt Ergebnisse unter Verwendung des folgenden Datensatzes und der folgenden Abfragen:

```
{ "created": "936864000", "dir_name": "important_docs", "files": [ { "name": "." },
  { "name": ".." }, { "name": ".aws" }, { "name": "downloads" } ], "owner": "Amazon
S3" }
{ "created": "936864000", "dir_name": "other_docs", "files": [ { "name": "." },
  { "name": ".." }, { "name": "my stuff" }, { "name": "backup" } ], "owner": "User" }
```

```
SELECT d.dir_name, d.files FROM S3object[*] d
```

```
{"dir_name":"important_docs","files":[{"name":"."}, {"name":".."}, {"name":".aws"},
{"name":"downloads"}]}
{"dir_name":"other_docs","files":[{"name":"."}, {"name":".."}, {"name":"my stuff"},
{"name":"backup"}]}
```

```
SELECT _1.dir_name, _1.owner FROM S3Object[*]
```

```
{"dir_name":"important_docs","owner":"Amazon S3"}  
{"dir_name":"other_docs","owner":"User"}
```

WHERE-Klausel

Die WHERE-Klausel hat die folgende Syntax:

```
WHERE condition
```

Die WHERE-Klausel filtert Zeilen basierend auf den Wert für „*condition*“. Eine Bedingung ist ein Ausdruck, der als Ergebnis einen booleschen Wert zurückgibt. Nur Zeilen, deren Bedingung als TRUE ausgewertet wird, werden als Ergebnis zurückgegeben.

LIMIT-Klausel

Die LIMIT-Klausel hat die folgende Syntax:

```
LIMIT number
```

Die LIMIT-Klausel begrenzt die Anzahl der von der Abfrage zurückgegebenen Datensätze basierend auf den Wert für „*number*“.

Attributzugriff

Die SELECT WHERE UND-Klauseln können mithilfe einer der Methoden in den folgenden Abschnitten auf Datensatzdaten verweisen, je nachdem, ob es sich bei der abgefragten Datei um ein oder um ein Format handelt. CSV JSON

CSV

- Spaltennummern — Sie können auf die N-te Spalte einer Zeile mit dem Spaltennamen verweisen, wobei *_NN* ist die Position der Spalte. Der Positionszähler beginnt mit 1. Die erste Spalte heißt demnach *_1*, die zweite Spalte heißt *_2*.

Sie können mit *_N* oder *alias*._*N* auf eine Spalte verweisen. Beispielsweise sind sowohl *_2* und *myAlias*._*2* gültige Möglichkeiten, um auf eine Spalte in der SELECT-Liste und der WHERE-Klausel zu verweisen.

- Spaltenüberschriften — Bei Objekten im CSV Format, die über eine Kopfzeile verfügen, sind die Überschriften für die SELECT Liste und WHERE die Klausel verfügbar. Insbesondere können Sie, wie bei herkömmlichen Ausdrücken SQL, innerhalb SELECT - und WHERE Klauselausdrücken, mit oder auf die Spalten verweisen. *alias.column_name column_name*

JSON

- Dokument — Sie können auf JSON Dokumentfelder zugreifen als *alias.name*. Auch der Zugriff auf verschachtelte Felder ist möglich, z. B *alias.name1.name2.name3*.
- Liste — Sie können auf Elemente in einer JSON Liste zugreifen, indem Sie nullbasierte Indizes mit dem Operator verwenden. [] Beispielsweise lässt sich das zweite Element einer Liste mithilfe von *alias[1]* aufrufen. Sie können den Zugriff auf Listenelemente mit dem Zugriff auf Felder kombinieren, z. B *alias.name1.name2[1].name3*.
- Beispiele: Betrachten Sie dieses JSON Objekt als Beispieldatensatz:

```
{
  "name": "Susan Smith",
  "org": "engineering",
  "projects":
    [
      {"project_name":"project1", "completed":false},
      {"project_name":"project2", "completed":true}
    ]
}
```

Beispiel 1

Die folgende Abfrage gibt die folgenden Ergebnisse zurück:

```
Select s.name from S3Object s
```

```
{"name":"Susan Smith"}
```

Beispiel 2

Die folgende Abfrage gibt die folgenden Ergebnisse zurück:

```
Select s.projects[0].project_name from S3Object s
```

```
{"project_name": "project1"}
```

Groß-/Kleinschreibung bei Header- und Attributnamen

Mit Amazon S3 Select können Sie doppelte Anführungszeichen verwenden, um anzugeben, dass bei Spaltenüberschriften (für CSV Objekte) und Attributen (für JSON Objekte) zwischen Groß- und Kleinschreibung unterschieden wird. Ohne doppelte Anführungszeichen muss die Groß-/Kleinschreibung bei Objekt-Headern und -attributen nicht berücksichtigt werden. Im Falle einer Zweideutigkeit wird ein Fehler ausgegeben.

Die folgenden Beispiele sind entweder 1) Amazon S3 S3-Objekte im CSV Format mit den angegebenen Spaltenüberschriften und mit der `FileHeaderInfo` Einstellung auf "Use" für die Abfrageanforderung oder 2) Amazon S3 S3-Objekte im JSON Format mit den angegebenen Attributen.

Beispiel 1: Das abzufragende Objekt hat den Header oder das Attribut „NAME“.

- Mit folgendem Ausdruck werden Werte von dem Objekt erfolgreich zurückgegeben. Da keine Anführungszeichen vorhanden sind, berücksichtigt die Abfrage die Groß- und Kleinschreibung nicht.

```
SELECT s.name from S3object s
```

- Der folgende Ausdruck führt zum Fehler 400 „MissingHeaderName“. Da Anführungszeichen vorhanden sind, berücksichtigt die Abfrage die Groß- und Kleinschreibung.

```
SELECT s."name" from S3object s
```

Beispiel 2: Das abzufragende Amazon S3-Objekt hat einen Header oder ein Attribut mit „NAME“ sowie einen anderen Header oder ein anderes Attribut mit „name“.

- Der folgende Ausdruck führt zum Fehler 400 „AmbiguousFieldName“. Da keine Anführungszeichen vorhanden sind, wird die Groß-/Kleinschreibung nicht beachtet, aber es gibt zwei Übereinstimmungen, weshalb der Fehler ausgegeben wird.

```
SELECT s.name from S3object s
```

- Mit folgendem Ausdruck werden Werte von dem Objekt erfolgreich zurückgegeben. Da Anführungszeichen vorhanden sind, berücksichtigt die Abfrage die Groß- und Kleinschreibung, weshalb es keine Mehrdeutigkeit gibt.

```
SELECT s."NAME" from S3object s
```

Verwenden von reservierten Schlüsselwörtern als benutzerdefinierte Begriffe

Amazon S3 Select verfügt über eine Reihe von reservierten Schlüsselwörtern, die zur Ausführung der SQL Ausdrücke benötigt werden, die zur Abfrage von Objektinhalten verwendet werden. Zu den reservierten Schlüsselwörtern zählen Funktionsnamen, Datentypen, Operatoren etc. In einigen Fällen können benutzerdefinierte Begriffe, wie die Spaltenüberschriften (für CSV Dateien) oder Attribute (für JSON Objekte), mit einem reservierten Schlüsselwort kollidieren. In diesem Fall geben Sie mithilfe von doppelten Anführungszeichen an, dass Sie absichtlich einen benutzerdefinierten Begriff verwenden, der mit einem reservierten Schlüsselwort übereinstimmt. Andernfalls wird ein 400-Parse-Fehler ausgegeben.

Eine vollständige Liste der reservierten Schlüsselwörter finden Sie unter [Reservierte Schlüsselwörter](#).

Das folgende Beispiel ist entweder 1) ein Amazon S3 S3-Objekt im CSV Format mit den angegebenen Spaltenüberschriften, das "Use" für die Abfrageanforderung auf `FileHeaderInfo` gesetzt ist, oder 2) ein Amazon S3 S3-Objekt im JSON Format mit den angegebenen Attributen.

Beispiel: Das abzufragende Objekt hat einen Header oder ein Attribut mit dem Namen „CAST“, der ein reserviertes Schlüsselwort ist.

- Mit folgendem Ausdruck werden Werte von dem Objekt erfolgreich zurückgegeben. Da die Abfrage Anführungszeichen enthält, verwendet S3 Select den benutzerdefinierten Header oder das benutzerdefinierte Attribut.

```
SELECT s."CAST" from S3object s
```

- Der folgende Ausdruck führt zu einem Analysefehler 400. Da in der Abfrage keine Anführungszeichen verwendet werden, steht CAST im Konflikt mit einem reservierten Schlüsselwort.

```
SELECT s.CAST from S3object s
```

Skalare Ausdrücke

In der WHERE Klausel und der SELECT Liste können Sie SQL skalare Ausdrücke verwenden, d. h. Ausdrücke, die Skalarwerte zurückgeben. Sie haben das folgende Format:

- ***Literal***

Ein SQL Literal.

- ***column_reference***

Ein Verweis auf eine Spalte in der Form *column_name* oder *alias.column_name*.

- ***unary_op expression***

In diesem Fall ***unary_op*** ist ein unärer OperatorSQL.

- ***expression binary_op expression***

In diesem Fall ***binary_op*** ist es ein SQL binärer Operator.

- ***func_name***

In diesem Fall ist ***func_name*** der Name der aufzurufenden skalaren Funktion.

- ***expression*** [NOT] BETWEEN ***Ausdruck*** AND ***Ausdruck***

- ***expression*** LIKE ***Ausdruck*** [ESCAPE ***expression***]

Datentypen

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt mehrere primitive Datentypen.


Datentypkonvertierungen

Allgemein sollte die Funktion CAST verwendet werden, wenn sie definiert ist. Falls CAST nicht definiert ist, werden alle Eingabedaten als Zeichenfolge betrachtet. In diesem Fall müssen Sie Ihre Eingabedaten bei Bedarf in die relevanten Datentypen umwandeln.

Weitere Informationen zur Funktion CAST finden Sie unter [CAST](#).

Unterstützte Datentypen

Amazon S3 Select unterstützt die folgenden primitiven Datentypen.

Name	Beschreibung	Beispiele
<code>bool</code>	Ein boolescher Wert, entweder TRUE oder FALSE.	FALSE
<code>int, integer</code>	Eine 8-Byte-Ganzzahl im Bereich -9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807.	100000
<code>string</code>	Eine UTF8 -kodierte Zeichenfolge mit variabler Länge. Das Standardlimit ist 1 Zeichen. Das maximale Zeichenlimit beträgt 2.147.483.647.	'xyz'
<code>float</code>	Eine 8-Byte-Gleitkommazahl.	CAST(0.456 AS FLOAT)
<code>decimal, numeric</code>	Eine Zahl mit der Basis 10 mit einer maximalen Genauigkeit von 38 (d. h. maximale Anzahl signifikanter Ziffern) und einer Größe zwischen -2^{31} und $2^{31}-1$ (d. h. der Exponent der Basis 10).	123.456
<div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>Amazon S3 Select ignoriert die Skalierung und Genauigkeit, wenn Sie beide gleichzeitig bereitstellen.</p> </div>		
<code>timestamp</code>	Zeitstempel stellen einen bestimmten Zeitpunkt dar, enthalten immer einen lokalen Versatz und können beliebig genau sein. Im Textformat befolgen Zeitstempel den W3C-Hinweis zu Datums- und Zeitformaten , müssen jedoch mit dem Buchstaben „T“ enden, wenn die Präzision der Zeitstempel	CAST('2007-04-05T14:30Z' AS TIMESTAMP)

Name	Beschreibung	Beispiele
	nicht mindestens ein ganzer Tag ist. Sekundenbruchteile mit mindestens einer Stelle bis zu beliebig vielen Stellen sind zulässig. Offsets zur Ortszeit können entweder als Zeitversatz von Stunde zu Minute oder als Literalwert für die Ortszeit von UTC dargestellt werden. Z UTC Lokale Zeitversätze sind für Zeitstempel mit Uhrzeit erforderlich, für reine Datumswerte jedoch nicht zulässig.	

Unterstützte Parquet-Typen

Amazon S3 Select unterstützt die folgenden Parquet-Typen.

- DATE
- DECIMAL
- ENUM
- INT(8)
- INT(16)
- INT(32)
- INT(64)
- LIST

Note

Für die LIST Parquet Typausgabe unterstützt Amazon S3 Select nur das JSON Format. Wenn die Abfrage die Daten jedoch auf einfache Werte beschränkt, kann der LIST Parquet Typ auch im CSV Format abgefragt werden.

- STRING
- TIMESTAMP-unterstützte Präzision (MILLIS/MICROS/NANOS)

Note

Zeitstempel, die als INT(96) gespeichert werden, werden nicht unterstützt.

Aufgrund des Bereichs des Typs `INT(64)` können Zeitstempel in der Einheit `NANOS` nur Werte zwischen `1677-09-21 00:12:43` und `2262-04-11 23:47:16` darstellen. Werte, die außerhalb dieses Bereichs liegen, können mit der `NANOS`-Einheit nicht dargestellt werden.

Zuordnung von Parquet-Typen zu unterstützten Datentypen in Amazon S3 Select

Parquet-Typen	Unterstützte Datentypen
<code>DATE</code>	<code>timestamp</code>
<code>DECIMAL</code>	<code>decimal, numeric</code>
<code>ENUM</code>	<code>string</code>
<code>INT(8)</code>	<code>int, integer</code>
<code>INT(16)</code>	<code>int, integer</code>
<code>INT(32)</code>	<code>int, integer</code>
<code>INT(64)</code>	<code>decimal, numeric</code>
<code>LIST</code>	Jeder Parquet-Typ in der Liste wird auf den entsprechenden Datentyp abgebildet.
<code>STRING</code>	<code>string</code>
<code>TIMESTAMP</code>	<code>timestamp</code>

Operatoren

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt die folgenden Operatoren.

Logische Operatoren

- AND
- NOT
- OR

Vergleichsoperatoren

- <
- >
- <=
- >=
- =
- <>
- !=
- BETWEEN
- IN – Beispiel: IN ('a', 'b', 'c')

Mustervergleichsoperatoren

- LIKE
- _ (Vergleicht ein beliebiges Zeichen)
- % (Vergleicht eine beliebige Zeichenreihenfolge)

Einheitliche Operatoren

- IS NULL
- IS NOT NULL

Mathematische Operatoren

Es werden Addition, Subtraktion, Multiplikation, Division und Modulo wie folgt unterstützt:

- +
- -
- *
- /
- %

Rangfolge der Operatoren

Die folgende Tabelle zeigt die Rangfolge der Operatoren in absteigender Reihenfolge.

Operator oder Element	Assoziativität	Erforderlich
-	rechts	unär minus
*, /, %	links	Multiplikation, Division und Modulo
+, -	links	Addition und Subtraktion
IN		Mitgliedschaft festlegen
BETWEEN		in Bereich enthalten

Operator oder Element	Assoziativität	Erforderlich
LIKE		Zeichenfolgenübereinstimmung
<>		kleiner als, größer als
=	rechts	Gleichheit, Zuweisung
NOT	rechts	logische Negation
AND	links	logische Verbindung
OR	links	logische Disjunktion

Reservierte Schlüsselwörter

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Nachfolgend finden Sie eine Liste der reservierten Schlüsselwörter für Amazon S3 Select. Zu diesen Schlüsselwörtern gehören Funktionsnamen, Datentypen, Operatoren usw., die zur Ausführung der SQL Ausdrücke benötigt werden, die zur Abfrage von Objekten verwendet werden.

```
absolute
action
add
all
allocate
```

alter
and
any
are
as
asc
assertion
at
authorization
avg
bag
begin
between
bit
bit_length
blob
bool
boolean
both
by
cascade
cascaded
case
cast
catalog
char
char_length
character
character_length
check
clob
close
coalesce
collate
collation
column
commit
connect
connection
constraint
constraints
continue
convert
corresponding

count
create
cross
current
current_date
current_time
current_timestamp
current_user
cursor
date
day
deallocate
dec
decimal
declare
default
deferrable
deferred
delete
desc
describe
descriptor
diagnostics
disconnect
distinct
domain
double
drop
else
end
end-exec
escape
except
exception
exec
execute
exists
external
extract
false
fetch
first
float
for

foreign
found
from
full
get
global
go
goto
grant
group
having
hour
identity
immediate
in
indicator
initially
inner
input
insensitive
insert
int
integer
intersect
interval
into
is
isolation
join
key
language
last
leading
left
level
like
limit
list
local
lower
match
max
min
minute

missing
module
month
names
national
natural
nchar
next
no
not
null
nullif
numeric
octet_length
of
on
only
open
option
or
order
outer
output
overlaps
pad
partial
pivot
position
precision
prepare
preserve
primary
prior
privileges
procedure
public
read
real
references
relative
restrict
revoke
right
rollback

rows
schema
scroll
second
section
select
session
session_user
set
sexp
size
smallint
some
space
sql
sqlcode
sqlerror
sqlstate
string
struct
substring
sum
symbol
system_user
table
temporary
then
time
timestamp
timezone_hour
timezone_minute
to
trailing
transaction
translate
translation
trim
true
tuple
union
unique
unknown
unpivot
update

```
upper
usage
user
using
value
values
varchar
varying
view
when
whenever
where
with
work
write
year
zone
```

SQLFunktionen

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt die folgenden SQL Funktionen.

Themen

- [Aggregationsfunktionen](#)
- [Konditionale Funktionen](#)
- [Konvertierungs-Funktionen](#)
- [Datumsfunktionen](#)
- [Zeichenfolgenfunktionen](#)

Aggregationsfunktionen

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt die folgenden aggregierten Funktionen.

Funktion	Argumenttyp	Rückgabetyt
AVG(<i>expressic</i> <i>n</i>)	INT, FLOAT, DECIMAL	DECIMAL für ein INT-Argument, FLOAT für ein Gleitkomm a-Argumen t; andernfal ls gleich dem Argumentd atentyp.
COUNT	-	INT
MAX(<i>expressic</i> <i>n</i>)	INT, DECIMAL	Entspricht dem Argumenttyp.
MIN(<i>expressic</i> <i>n</i>)	INT, DECIMAL	Entspricht dem Argumenttyp.
SUM(<i>expressic</i> <i>n</i>)	INT, FLOAT, DOUBLE, DECIMAL	INT für ein INT-Argument, FLOAT für ein Gleitkomm a-Argumen t; andernfal

Funktion	Argumenttyp	Rückgabotyp
		Is gleich dem Argumentd atentyp.

SUMBeispiel für

Verwenden Sie einen SUM-Ausdruck, um die gesamten Objektgrößen eines Ordners in einem [S3-Bestandslistenbericht](#) zusammenzufassen.

Der folgende S3-Inventarbericht ist eine CSV Datei, die mit komprimiert wurdeGZIP. Sie hat drei Spalten.

- Die erste Spalte enthält den Namen des S3-Buckets (*DOC-EXAMPLE-BUCKET*), für den der S3-Inventarbericht bestimmt ist.
- Die zweite Spalte enthält den Objektschlüsselnamen, der das Objekt eindeutig im Bucket identifiziert.

Der *example-folder/-*-Wert in der ersten Zeile steht für den Ordner *example-folder*. Wenn Sie in Amazon S3 einen Ordner in Ihrem Bucket anlegen, erstellt S3 ein 0-Byte-Objekt mit einem Schlüssel, der auf den von Ihnen angegebenen Ordernamen festgelegt ist.

Der *example-folder/object1*-Wert in der zweiten Zeile steht für das Objekt *object1* im Ordner *example-folder*.

Der *example-folder/object2*-Wert in der dritten Zeile steht für das Objekt *object2* im Ordner *example-folder*.

Weitere Informationen über S3-Ordner finden Sie unter [Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern](#).

- Die dritte Spalte enthält die Objektgröße in Byte.

```
"DOC-EXAMPLE-BUCKET", "example-folder/", "0"
"DOC-EXAMPLE-BUCKET", "example-folder/object1", "2011267"
"DOC-EXAMPLE-BUCKET", "example-folder/object2", "1570024"
```

Um einen SUM Ausdruck zur Berechnung der Gesamtgröße des Ordners zu verwenden *example-folder*, führen Sie die SQL Abfrage mit Amazon S3 Select aus.

```
SELECT SUM(CAST(_3 as INT)) FROM s3object s WHERE _2 LIKE 'example-folder/%' AND _2 != 'example-folder/';
```

Abfrageergebnis:

```
3581291
```

Konditionale Funktionen

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt die folgenden bedingten Funktionen.

Themen

- [CASE](#)
- [COALESCE](#)
- [NULLIF](#)

CASE

Der CASE-Ausdruck ist ein bedingter Ausdruck, der sich mit den `if/then/else`-Anweisungen anderer Sprachen vergleichen lässt. CASE wird verwendet, um ein Ergebnis anzugeben, wenn es mehrere Bedingungen gibt. Es gibt zwei Arten von CASE-Ausdrücken: einfach und gesucht.

In einfachen CASE-Ausdrücken wird ein Ausdruck mit einem Wert verglichen. Wenn keine Übereinstimmung gefunden wird, wird die in der THEN-Klausel angegebene Aktion angewendet. Wenn keine Übereinstimmung gefunden wird, wird die in der ELSE-Klausel angegebene Aktion angewendet.

In gesuchten CASE-Ausdrücken wird jeder CASE-Ausdruck auf der Basis eines booleschen Ausdrucks evaluiert und die CASE-Anweisung gibt den ersten übereinstimmenden CASE-Ausdruck zurück. Wenn

in den WHEN-Klauseln kein übereinstimmender CASE-Ausdruck gefunden wird, wird die Aktion in der ELSE-Klausel zurückgegeben.

Syntax

Note

Derzeit unterstützt Amazon S3 Select weder ORDER BY noch Abfragen, die neue Zeilen enthalten. Achten Sie darauf, Abfragen ohne Zeilenumbrüche zu verwenden.

Die folgende Zeichenfolge ist eine einfache CASE-Aussage, die verwendet wird, um Bedingungen abzugleichen:

```
CASE expression WHEN value THEN result [WHEN... ] [ELSE result] END
```

Die folgende Zeichenfolge ist eine gesuchte CASE-Anweisung, die verwendet wird, um jede Bedingung auszuwerten:

```
CASE WHEN boolean condition THEN result [WHEN ... ] [ELSE result] END
```

Beispiele

Note

Wenn Sie die folgenden Beispiele mit der Amazon S3 S3-Konsole ausführen und Ihre CSV Datei eine Kopfzeile enthält, wählen Sie Die erste CSV Datenzeile ausschließen.

Beispiel 1: Verwenden Sie einen einfachen CASE-Ausdruck, um New York City in einer Abfrage durch Big Apple zu ersetzen. Alle anderen Städtenamen werden durch other ersetzt.

```
SELECT venuecity, CASE venuecity WHEN 'New York City' THEN 'Big Apple' ELSE 'other' END  
FROM S3object;
```

Abfrageergebnis:

```
venuecity      | case  
-----+-----  
Los Angeles   | other
```



```
New York City | Big Apple
San Francisco | other
Baltimore     | other
...
```

Beispiel 2: Verwenden Sie einen gesuchten CASE-Ausdruck, um Gruppennummern basierend auf dem `pricepaid`-Wert für einzelne Ticketverkäufe zuzuweisen:

```
SELECT pricepaid, CASE WHEN CAST(pricepaid as FLOAT) < 10000 THEN 'group 1' WHEN
CAST(pricepaid as FLOAT) > 10000 THEN 'group 2' ELSE 'group 3' END FROM S3Object;
```

Abfrageergebnis:

```
pricepaid | case
-----+-----
12624.00 | group 2
10000.00 | group 3
10000.00 | group 3
9996.00  | group 1
9988.00  | group 1
...
```

COALESCE

COALESCE wertet die Argumente nacheinander aus und gibt den ersten unbekanntem Wert zurück, das heißt, den ersten Wert, der nicht null oder nicht fehlend ist. Null- und fehlende Werte werden von der Funktion nicht übernommen.

Syntax

```
COALESCE ( expression, expression, ... )
```

Parameter

expression

Der Zielausdruck, der von der Funktion verwendet wird.

Beispiele

```
COALESCE(1)          -- 1
```

```
COALESCE(null)           -- null
COALESCE(null, null)    -- null
COALESCE(missing)       -- null
COALESCE(missing, missing) -- null
COALESCE(1, null)       -- 1
COALESCE(null, null, 1) -- 1
COALESCE(null, 'string') -- 'string'
COALESCE(missing, 1)    -- 1
```

NULLIF

Bei zwei Ausdrücken, die das gleiche Auswertungsergebnis haben, gibt NULLIF NULL zurück. Andernfalls gibt NULLIF das Auswertungsergebnis für den ersten Ausdruck zurück.

Syntax

```
NULLIF ( expression1, expression2 )
```

Parameter

expression1, *expression2*

Die Zielausdrücke, die von der Funktion verwendet werden.

Beispiele

```
NULLIF(1, 1)           -- null
NULLIF(1, 2)           -- 1
NULLIF(1.0, 1)         -- null
NULLIF(1, '1')         -- 1
NULLIF([1], [1])       -- null
NULLIF(1, NULL)        -- 1
NULLIF(NULL, 1)        -- null
NULLIF(null, null)     -- null
NULLIF(missing, null)  -- null
NULLIF(missing, missing) -- null
```

Konvertierungs-Funktionen

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt die folgende Konvertierungsfunktion.

Themen

- [CAST](#)

CAST

Die CAST-Funktion konvertiert ein Element, z. B. einen Ausdruck zur Auswertung eines einzelnen Werts, von einem Typ in einen anderen.

Syntax

```
CAST ( expression AS data_type )
```

Parameter

expression

Eine Kombination aus einem oder mehreren Werten, Operatoren und SQL Funktionen, die einen Wert ergeben.

data_type

Der Zieldatentyp, z. B. INT, in den der Ausdruck umgewandelt werden soll Eine Liste der unterstützten Datentypen finden Sie unter [Datentypen](#).

Beispiele

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)  
CAST(0.456 AS FLOAT)
```

Datumsfunktionen

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt die folgenden Datumsfunktionen.

Themen

- [DATE_ADD](#)
- [DATE_DIFF](#)
- [EXTRACT](#)
- [TO_STRING](#)
- [TO_TIMESTAMP](#)
- [UTCNOW](#)

DATE_ADD

Bei einem Datumsteil, einer Menge und einem Zeitstempel gibt DATE_ADD einen aktualisierten Zeitstempel zurück, indem der Datumsteil anhand der Menge modifiziert wird.

Syntax

```
DATE_ADD( date_part, quantity, timestamp )
```

Parameter

date_part

Gibt den zu modifizierenden Teil des Datums an. Dabei kann es sich um einen der folgenden Werte handeln:

- Jahr
- Monat
- Tag

- Stunde
- Minute
- Sekunde

quantity

Der Wert, der auf den aktualisierten Zeitstempel anzuwenden ist. Positive „*quantity*“-Werte werden zum „*date_part*“-Wert des Zeitstempels addiert, negative Werte werden subtrahiert.

timestamp

Der Zielzeitstempel, der von der Funktion verwendet wird.

Beispiele

```
DATE_ADD(year, 5, `2010-01-01T`)           -- 2015-01-01 (equivalent to
  2015-01-01T)
DATE_ADD(month, 1, `2010T`)                 -- 2010-02T (result will add precision
  as necessary)
DATE_ADD(month, 13, `2010T`)                -- 2011-02T
DATE_ADD(day, -1, `2017-01-10T`)           -- 2017-01-09 (equivalent to
  2017-01-09T)
DATE_ADD(hour, 1, `2017T`)                  -- 2017-01-01T01:00-00:00
DATE_ADD(hour, 1, `2017-01-02T03:04Z`)     -- 2017-01-02T04:04Z
DATE_ADD(minute, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:05:05.006Z
DATE_ADD(second, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:04:06.006Z
```

DATE_DIFF

Bei einem Datumsteil und zwei gültigen Zeitstempeln gibt DATE_DIFF die Differenz in Datumsteilen an. Sofern der *date_part*-Wert von *timestamp1* größer ist als der *date_part*-Wert von *timestamp2*, wird eine negative Ganzzahl zurückgegeben. Wenn der *date_part*-Wert von *timestamp1* kleiner ist als der *date_part*-Wert von *timestamp2*, wird eine positive Ganzzahl zurückgegeben.

Syntax

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

Parameter

date_part

Gibt den zu vergleichenden Teil der Zeitstempel an. Die Definition von *date_part* finden Sie unter [DATE_ADD](#).

timestamp1

Der erste Zeitstempel für den Vergleich.

timestamp2

Der zweite Zeitstempel für den Vergleich.

Beispiele

```
DATE_DIFF(year, `2010-01-01T`, `2011-01-01T`)           -- 1
DATE_DIFF(year, `2010T`, `2010-05T`)                   -- 4 (2010T is equivalent to
2010-01-01T00:00:00.000Z)
DATE_DIFF(month, `2010T`, `2011T`)                     -- 12
DATE_DIFF(month, `2011T`, `2010T`)                     -- -12
DATE_DIFF(day, `2010-01-01T23:00`, `2010-01-02T01:00`) -- 0 (need to be at least 24h
apart to be 1 day apart)
```

EXTRACT

Bei einem Datumsteil und einem Zeitstempel gibt EXTRACT den Datumsteilwert des Zeitstempels zurück.

Syntax

```
EXTRACT( date_part FROM timestamp )
```

Parameter

date_part

Gibt den zu extrahierenden Teil der Zeitstempel an. Dabei kann es sich um einen der folgenden Werte handeln:

- YEAR
- MONTH

- DAY
- HOUR
- MINUTE
- SECOND
- TIMEZONE_HOUR
- TIMEZONE_MINUTE

Zeitstempel

Der Zielzeitstempel, der von der Funktion verwendet wird.

Beispiele

```
EXTRACT(YEAR FROM `2010-01-01T`)           -- 2010
EXTRACT(MONTH FROM `2010T`)                -- 1 (equivalent to
2010-01-01T00:00:00.000Z)
EXTRACT(MONTH FROM `2010-10T`)             -- 10
EXTRACT(HOUR FROM `2017-01-02T03:04:05+07:08`) -- 3
EXTRACT(MINUTE FROM `2017-01-02T03:04:05+07:08`) -- 4
EXTRACT(TIMEZONE_HOUR FROM `2017-01-02T03:04:05+07:08`) -- 7
EXTRACT(TIMEZONE_MINUTE FROM `2017-01-02T03:04:05+07:08`) -- 8
```

TO_STRING

Bei einem Zeitstempel und einem Formatmuster gibt TO_STRING eine Zeichenfolgendarstellung des Zeitstempels im angegebenen Format zurück.

Syntax

```
TO_STRING ( timestamp time_format_pattern )
```

Parameter

timestamp

Der Zielzeitstempel, der von der Funktion verwendet wird.

time_format_pattern

Eine Zeichenfolge mit folgenden speziellen Zeichenbedeutungen:

Format	Beispiel	Beschreibung
yy	69	Jahreszahl mit 2 Ziffern
y	1969	Jahreszahl mit 4 Ziffern
yyyy	1969	Jahreszahl mit 4 Ziffern, mit Nullen aufgefüllt
M	1	Monatsname
MM	01	Monatsname, mit Nullen aufgefüllt
MMM	Jan	Abkürzung des Monatsnamens
MMMM	January	Vollständiger Monatsnamen
MMMMM	J	Erster Buchstabe des Monats des Jahres (NOTE: Dieses Format ist für die Verwendung mit der TO_TIMESTAMP

Format	Beispiel	Beschreibung
		AMP Funktion nicht gültig.)
d	2	Monatstag (1-31)
dd	02	Monatstag , mit Nullen aufgefüllt (01-31)
a	AM	AM oder PM
h	3	Stunde (1-12)
hh	03	Stunde, mit Nullen aufgefüllt (01-12)
H	3	Stunde (0-23)
HH	03	Stunde, mit Nullen aufgefüllt (00-23)
m	4	Minute (0-59)
mm	04	Minute, mit Nullen aufgefüllt (00-59)
s	5	Sekunde (0-59)

Format	Beispiel	Beschreibung
ss	05	Sekunde, mit Nullen aufgefüllt (00-59)
S	0	Sekundenbruchteil (Genauigkeit: 0,1, Bereich: 0,0-0,9)
SS	6	Sekundenbruchteil (Genauigkeit: 0,01, Bereich: 0,0-0,99)
SSS	60	Sekundenbruchteil (Genauigkeit: 0,001, Bereich: 0,0-0,999)
...
SSSSSSSSS	60000000	Sekundenbruchteil (max. Genauigkeit: 1 Nanosekunde, Bereich: 0,0-0,99999999)
n	60000000	Nanosekunde

Format	Beispiel	Beschreibung
X	+07 oder Z	Offset in Stunden oder „Z“ bei Offset = 0
XX oder XXXX	+0700 oder Z	Offset in Stunden und Minuten oder „Z“ bei Offset = 0
XXX oder XXXXX	+07:00 oder Z	Offset in Stunden und Minuten oder „Z“ bei Offset = 0
x	7	Offset in Stunden
xx oder xxxx	700	Offset in Stunden und Minuten
xxx oder xxxxx	+07:00	Offset in Stunden und Minuten

Beispiele

```

TO_STRING(`1969-07-20T20:18Z`, 'MMMM d, y')           -- "July 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMM d, yyyy')        -- "Jul 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'M-d-yy')            -- "7-20-69"
TO_STRING(`1969-07-20T20:18Z`, 'MM-d-y')            -- "07-20-1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMMM d, y h:m a')   -- "July 20, 1969 8:18
PM"

```

```
TO_STRING(`1969-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX') --
"1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX') --
"1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX') --
"1969-07-20T20:18:00+0800"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXXX') --
"1969-07-20T20:18:00+08:00"
```

TO_TIMESTAMP

Bei einer Zeichenfolge wandelt TO_TIMESTAMP diese in einen Zeitstempel um. TO_TIMESTAMP ist der umgekehrte Vorgang von TO_STRING.

Syntax

```
TO_TIMESTAMP ( string )
```

Parameter

string

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
TO_TIMESTAMP('2007T') -- `2007T`
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

UTCNOW

UTCNOW gibt die aktuelle Uhrzeit UTC als Zeitstempel zurück.

Syntax

```
UTCNOW()
```

Parameter

UTCNOW nutzt keine Parameter.

Beispiele

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

Zeichenfolgenfunktionen

Important

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#)

Amazon S3 Select unterstützt die folgenden String-Funktionen.

Themen

- [CHAR_LENGTH, CHARACTER_LENGTH](#)
- [LOWER](#)
- [SUBSTRING](#)
- [TRIM](#)
- [UPPER](#)

CHAR_LENGTH, CHARACTER_LENGTH

CHAR_LENGTH (oder CHARACTER_LENGTH) zählt die Anzahl der Zeichen in der angegebenen Zeichenfolge.

Note

CHAR_LENGTH und CHARACTER_LENGTH sind Synonyme.

Syntax

```
CHAR_LENGTH ( string )
```

Parameter

string

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
CHAR_LENGTH('')          -- 0
CHAR_LENGTH('abcdefg')  -- 7
```

LOWER

LOWER wandelt alle Großbuchstaben einer Zeichenfolge in Kleinbuchstaben um. Alle Zeichen, die keine Großbuchstaben sind, bleiben unverändert.

Syntax

```
LOWER ( string )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
LOWER('AbCdEfG!@#') -- 'abcdefg!@#'
```

SUBSTRING

SUBSTRING gibt bei einer Zeichenfolge, einem Startindex und (optional) einer Länge die Teilzeichenfolge vom Startindex bis zum Ende der Zeichenfolge oder bis zur angegebenen Länge zurück.

Note

Das erste Zeichen der Eingabezeichenfolge hat die Indexposition 1.

- Wenn $start < 1$ ist und keine Länge angegeben wird, dann wird die Indexposition auf 1 festgelegt.
- Wenn $start < 1$ ist und eine Länge angegeben wird, dann wird die Indexposition auf $start + length - 1$ festgelegt.
- Wenn $start + length - 1 < 0$ ist, dann wird eine leere Zeichenfolge zurückgegeben.
- Wenn $start + length - 1 \geq 0$ ist, dann wird die Teilzeichenfolge beginnend bei Indexposition 1 mit der Länge $start + length - 1$ zurückgegeben.

Syntax

```
SUBSTRING( string FROM start [ FOR length ] )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

start

Die Startposition der Zeichenfolge.

length

Die Länge der zurückzugebenden Teilzeichenfolge. Falls nicht angegeben, wird bis zum Ende der Zeichenfolge fortgefahren.

Beispiele

```
SUBSTRING("123456789", 0)      -- "123456789"  
SUBSTRING("123456789", 1)     -- "123456789"  
SUBSTRING("123456789", 2)     -- "23456789"  
SUBSTRING("123456789", -4)    -- "123456789"  
SUBSTRING("123456789", 0, 999) -- "123456789"  
SUBSTRING("123456789", 1, 5)  -- "12345"
```

TRIM

Kürzt vorangestellte oder nachgestellte Zeichen aus einer Zeichenfolge. Standardmäßig wird ein Leerzeichen (' ') entfernt.

Syntax

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

LEADING | TRAILING | BOTH

Dieser Parameter gibt an, ob vorangestellte oder nachgestellte Zeichen oder vorangestellte und nachgestellte Zeichen gekürzt werden sollen.

remove_chars

Die zu entfernenden Zeichen. *remove_chars* kann eine Zeichenfolge mit einer Länge von > 1 sein. Diese Funktion gibt die Zeichenfolge mit einem beliebigen Zeichen von *remove_chars* zurück, der am Anfang oder am Ende der entfernten Zeichenfolge stand.

Beispiele

```
TRIM('   foobar   ') -- 'foobar'  
TRIM('   \tfoobar\t   ') -- '\tfoobar\t'  
TRIM(LEADING FROM '   foobar   ') -- 'foobar'  
TRIM(TRAILING FROM '   foobar   ') -- '   foobar'  
TRIM(BOTH FROM '   foobar   ') -- 'foobar'  
TRIM(BOTH '12' FROM '1112211foobar22211122') -- 'foobar'
```

UPPER

UPPER wandelt alle Kleinbuchstaben einer Zeichenfolge in Großbuchstaben um. Alle Zeichen, die keine Kleinbuchstaben sind, bleiben unverändert.

Syntax

```
UPPER ( string )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
UPPER('AbCdEfG!@#') -- 'ABCDEFGH!@#'
```

Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch.

Sie können S3-Batch-Vorgänge verwenden, um umfangreiche Batch-Vorgänge für Amazon S3-Objekte durchzuführen. S3-Batch-Vorgänge kann eine einzelne Operation für Listen von Amazon S3-Objekten durchführen, die Sie angeben. Ein einziger Auftrag kann eine festgelegte Operation auf Milliarden von Objekten mit mehreren Exabytes an Daten durchführen. Amazon S3 verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. So profitieren Sie von einer vollständig verwalteten, prüfbar und serverlosen Umgebung. Sie können S3-Batch-Vorgänge über die AWS Management Console, die AWS CLI, Amazon-SDKs oder die REST-API verwenden.

Verwenden Sie S3-BatchVorgänge, um Objekte zu kopieren und Objekt-Markierungen oder Zugriffskontrolllisten (ACLs) festzulegen. Sie können Objekt-Wiederherstellungen auch von S3 Glacier Flexible Retrieval aus initiieren oder eine AWS Lambda-Funktion aufrufen, um benutzerdefinierte Aktionen mit Ihren Objekten durchzuführen. Sie können diese Vorgänge für eine benutzerdefinierte Liste von Objekten durchführen oder mit einem Amazon S3 Inventory einfach Objektlisten erzeugen. Amazon S3-Batch-Vorgänge verwendet die gleichen Amazon S3-APIs, die Sie bereits mit Amazon S3 verwenden, sodass Ihnen die Benutzeroberfläche bereits vertraut ist.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-](#)

[Buckets](#). Weitere Informationen zur Verwendung von Batch Operations mit S3 Express One Zone und Verzeichnis-Buckets finden Sie unter [Verwenden von Batch Operations mit S3 Express One Zone](#).

Grundlagen von S3-Batchvorgänge

Sie können S3-Batch-Vorgänge verwenden, um umfangreiche Batch-Vorgänge für Amazon S3-Objekte durchzuführen. S3-Batch-Vorgänge kann eine einzelne Operation oder Aktion in Listen von Amazon S3-Objekten ausführen, die Sie angeben.

Terminologie

In diesem Abschnitt werden die Begriffe Aufträge, Vorgänge und Aufgaben erwähnt, ihre Bedeutung ist wie folgt zu verstehen:

Aufgabe

Ein Auftrag ist die grundlegende Arbeitseinheit für S3-Batchvorgänge. Eine Aufgabe enthält alle Informationen, die erforderlich sind, um die angegebene Operation für die im Manifest aufgeführten Objekte auszuführen. Sobald Sie diese Informationen bereitgestellt und den Beginn des Auftrags angefordert haben, führt die Aufgabe die Operation für alle Objekte im Manifest durch.

Operation

Die Operation stellt die Art der API-[Aktion](#) dar, z. B. das Kopieren von Objekten, die von dem Batchoperations-Auftrag ausgeführt werden soll. Jeder Auftrag führt einen einzelnen Typ von Operation für alle Objekte aus, die im Manifest angegeben sind.

Aufgabe

Eine Aufgabe ist die Ausführungseinheit für einen Auftrag. Eine Aufgabe steht für einen einzelnen Aufruf einer Amazon S3- oder AWS Lambda-API-Operation zur Durchführung der Auftragsoperation für ein einzelnes Objekt. Während der Lebensdauer eines Auftrag erstellt S3-Batch-Vorgänge eine Aufgabe für jedes Objekt, das im Manifest angegeben ist.

Funktionsweise eines S3-Batchoperations-Auftrags

Ein Auftrag ist die grundlegende Arbeitseinheit für S3-Batchvorgänge. Eine Aufgabe enthält alle Informationen, die erforderlich sind, um die angegebene Operation für eine Liste von Objekten

auszuführen. Um einen Auftrag zu erstellen, übergeben Sie S3-Batch-Vorgänge eine Liste von Objekten und geben die Aktion an, die für diese Objekte ausgeführt werden soll.

Informationen zu den Vorgängen, die S3 Batch Operations unterstützt, finden Sie unter [Von S3 Batch-Vorgänge unterstützte Vorgänge](#).

Ein Batch-Auftrag führt die festgelegte Operation für jedes Objekt aus, das im Manifest enthalten ist. Ein Manifest listet die Objekte auf, die ein Batch-Auftrag verarbeiten soll, und es wird als Objekt in einem Bucket gespeichert. Sie können einen kommagetrennten Wert (CSV)-formatierten [Amazon S3 Inventory](#)-Bericht als Manifest nutzen, was die Erstellung umfassender Listen von Objekten in einem Bucket erleichtert. Ein Manifest können Sie auch in einem einfachen CSV-Format festlegen, mit dem Sie Batch-Vorgänge für eine benutzerdefinierte Liste an Objekten durchführen können, die in einem einzelnen Bucket enthalten sind.

Nachdem Sie einen Auftrag erstellt haben, verarbeitet Amazon S3 die Liste der Objekte in dem Manifest und führt die festgelegte Operation für jedes Objekt aus. Während ein Auftrag ausgeführt wird, können Sie den Fortschritt programmgesteuert oder über die Amazon S3-Konsole überwachen. Sie können einen Auftrag auch so konfigurieren, dass er nach Ende der Ausführung einen Abschlussbericht erzeugt. Der Abschlussbericht beschreibt die Ergebnisse jeder Aufgabe, die von dem Auftrag durchgeführt wurde. Weitere allgemeine Informationen zur Überwachung von Aufträgen finden Sie unter [Verwalten von S3-Batch-Vorgangsaufträgen](#).

Tutorial zu S3-Batchvorgängen

Im folgenden Tutorial werden vollständige end-to-end Verfahren für einige Batchoperationenaufgaben vorgestellt.

- [Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations, AWS Lambda, und AWS Elemental MediaConvert](#)

Erteilen von Berechtigungen für Amazon-S3-BatchVorgänge

Bevor Sie S3 Batch Operation-Aufträge erstellen und ausführen, müssen Sie die erforderlichen Berechtigungen erteilen. Um einen Amazon-S3-Batch-Vorgangsauftrag zu erstellen, ist die `s3:CreateJob`-Benutzerberechtigung erforderlich. Dieselbe Entität, die den Job erstellt, muss auch die `iam:PassRole` Erlaubnis haben, die AWS Identity and Access Management (IAM) Rolle, die für den Job für Batch-Operationen angegeben ist.

Allgemeine Informationen zur Angabe von IAM Ressourcen finden Sie unter [IAMJSONRichtlinie, Ressourcenelemente](#) im IAMBenutzerhandbuch. Die folgenden Abschnitte enthalten Informationen zum Erstellen einer IAM Rolle und zum Anhängen von Richtlinien.

Themen

- [Eine IAM S3-Rolle für Batch Operations erstellen](#)
- [Anfügen von Berechtigungsrichtlinien](#)

Eine IAM S3-Rolle für Batch Operations erstellen

Amazon S3 benötigt Berechtigungen, um S3-Batch-Vorgänge für Sie auszuführen. Sie gewähren diese Berechtigungen über AWS Identity and Access Management (IAM) Rolle. Dieser Abschnitt enthält Beispiele für die Vertrauens- und Berechtigungsrichtlinien, die Sie beim Erstellen einer IAM Rolle verwenden. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [IAMRollen](#). Beispiele finden Sie unter [Steuern von Berechtigungen für S3-Batch-Vorgänge mithilfe von Auftrags-Markierungen](#) und [Kopieren von Objekten mit S3 BatchVorgänge](#).

In Ihren IAM Richtlinien können Sie auch Bedingungsschlüssel verwenden, um Zugriffsberechtigungen für S3 Batch Operations-Jobs zu filtern. Weitere Informationen und eine vollständige Liste der Amazon S3-spezifischen Bedingungsschlüssel finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Vertrauensrichtlinie

Damit der Dienstprinzipal von S3 Batch Operations die IAM Rolle übernehmen kann, fügen Sie der Rolle die folgende Vertrauensrichtlinie hinzu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batchoperations.s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
]
}
```

Anfügen von Berechtigungsrichtlinien

Je nach Art der Operation können Sie eine der folgenden Richtlinien anfügen.

Bevor Sie Berechtigungen konfigurieren, beachten Sie Folgendes:

- Unabhängig von der Operation benötigt Amazon S3 Berechtigungen, um das Manifestobjekt aus Ihrem S3-Bucket zu lesen und optional einen Bericht in Ihren Bucket zu schreiben. Daher enthalten alle der folgenden Richtlinien diese Berechtigungen.
- Für Amazon S3 S3-Inventarberichtsmanifeste benötigt S3 Batch Operations die Erlaubnis, das manifest.json-Objekt und alle zugehörigen CSV Datendateien zu lesen.
- Versionspezifische Berechtigungen wie `s3:GetObjectVersion` sind nur erforderlich, wenn Sie die Versions-ID der Objekte festlegen.
- Wenn Sie S3 Batch Operations für verschlüsselte Objekte ausführen, muss die IAM Rolle auch Zugriff auf die AWS KMS Schlüssel, die verwendet wurden, um sie zu verschlüsseln.
- Wenn Sie einen Inventarbericht einreichen, ist das Manifest verschlüsselt mit AWS KMS, muss Ihre IAM Richtlinie die Berechtigungen `"kms:Decrypt"` und `"kms:GenerateDataKey"` für das Objekt manifest.json und alle zugehörigen CSV Datendateien enthalten.
- Wenn der Batch Operations-Job ein Manifest in einem Bucket generiert, der ACLs aktiviert wurde und sich in einem anderen befindet AWS Konto, müssen Sie die `s3:PutObjectAcl` Berechtigung in der IAM Richtlinie der IAM Rolle erteilen, die für den Batch-Job konfiguriert ist. Wenn Sie diese Berechtigung nicht angeben, schlägt der Batch-Job mit der folgenden Fehlermeldung fehl `errorMessage occurred when preparing manifest: Failed to write manifest.`

Objekte kopieren: PutObject

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectTagging"
      ],

```

```

    "Effect": "Allow",
    "Resource": "arn:aws:s3:::DestinationBucket/*"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectAcl",
      "s3:GetObjectTagging",
      "s3:ListBucket"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:s3:::SourceBucket",
      "arn:aws:s3:::SourceBucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": [
      "arn:aws:s3:::ManifestBucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::ReportBucket/*"
    ]
  }
]
}

```

Objekt-Tagging ersetzen: PutObjectTagging

```

{
  "Version": "2012-10-17",
  "Statement": [

```

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObjectTagging",
    "s3:PutObjectVersionTagging"
  ],
  "Resource": "arn:aws:s3:::TargetResource/*"
},
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion"
  ],
  "Resource": [
    "arn:aws:s3:::ManifestBucket/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::ReportBucket/*"
  ]
}
]
```

Objekt-Tagging löschen: DeleteObjectTagging

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::TargetResource/*"
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": [
      "arn:aws:s3:::ManifestBucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::ReportBucket/*"
    ]
  }
]
}

```

Zugriffskontrollliste ersetzen: PutObjectAcl

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl"
      ],
      "Resource": "arn:aws:s3:::TargetResource/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
    }
  ]
}

```



```

    "Resource": [
      "arn:aws:s3:::ManifestBucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::ReportBucket/*"
    ]
  }
]
}

```

Objekte wiederherstellen: RestoreObject

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:RestoreObject"
      ],
      "Resource": "arn:aws:s3:::TargetResource/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],

```

```

    "Resource": [
      "arn:aws:s3:::ReportBucket/*"
    ]
  }
]
}

```

Beibehaltung von Object Lock anwenden: PutObjectRetention

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetBucketObjectLockConfiguration",
      "Resource": [
        "arn:aws:s3:::TargetResource"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectRetention",
        "s3:BypassGovernanceRetention"
      ],
      "Resource": [
        "arn:aws:s3:::TargetResource/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ]
    }
  ]
}

```

```

    ],
    "Resource": [
        "arn:aws:s3:::ReportBucket/*"
    ]
}
]
}

```

Rechtliche Sperre für Objektsperre anwenden: PutObjectLegalHold

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetBucketObjectLockConfiguration",
      "Resource": [
        "arn:aws:s3:::TargetResource"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObjectLegalHold",
      "Resource": [
        "arn:aws:s3:::TargetResource/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [

```

```

        "arn:aws:s3:::ReportBucket/*"
    ]
}
]
}

```

Replizieren Sie vorhandene Objekte: InitiateReplication mit einem von S3 generierten Manifest

Verwenden Sie diese Richtlinie, wenn Sie ein von S3 generiertes Manifest verwenden und speichern. Weitere Informationen zur Verwendung der Batchvorgänge zum Replizieren vorhandener Objekte finden Sie unter [Replizieren bestehender Objekte mit S3-Batch-Replikation](#).

```

{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Action":[
        "s3:InitiateReplication"
      ],
      "Effect":"Allow",
      "Resource":[
        "arn:aws:s3:::*** replication source bucket ***/*"
      ]
    },
    {
      "Action":[
        "s3:GetReplicationConfiguration",
        "s3:PutInventoryConfiguration"
      ],
      "Effect":"Allow",
      "Resource":[
        "arn:aws:s3:::*** replication source bucket ***"
      ]
    },
    {
      "Action":[
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Effect":"Allow",
      "Resource":[
        "arn:aws:s3:::*** manifest bucket ***/*"
      ]
    }
  ]
}

```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::*** completion report bucket ****/*",
        "arn:aws:s3:::*** manifest bucket ****/*"
      ]
    }
  ]
}

```

Replizieren Sie vorhandene Objekte: InitiateReplication mit einem Benutzermanifest

Verwenden Sie diese Richtlinie, wenn Sie ein vom Benutzer bereitgestelltes Manifest verwenden. Weitere Informationen zur Verwendung der Batchvorgänge zum Replizieren vorhandener Objekte finden Sie unter [Replizieren bestehender Objekte mit S3-Batch-Replikation](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:InitiateReplication"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::*** replication source bucket ***/*"
      ]
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::*** manifest bucket ***/*"
      ]
    }
  ],
}

```

```
    "Effect": "Allow",
    "Action": [
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::*** completion report bucket ***/*"
    ]
}
]
```

Erstellen eines S3-Batch-Vorgangsauftrags

Mit S3 Batch Operations können Sie umfangreiche Stapelvorgänge für eine Liste bestimmter Amazon-S3-Objekte durchführen. In diesem Abschnitt werden die Informationen beschrieben, die Sie zum Erstellen eines S3-Batch-Vorgangsauftrags benötigen, sowie die Ergebnisse einer CreateJob-Anforderung. Es enthält auch Anweisungen zum Erstellen eines Batch Operations-Jobs mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und AWS SDK for Java.

Wenn Sie einen S3-Batchvorgangsauftrag erstellen, können Sie einen Abschlussbericht für alle oder nur für die fehlgeschlagenen Aufträge anfordern. Sofern mindestens eine Aufgabe erfolgreich aufgerufen wurde, generiert S3 Batch Operations einen Bericht für abgeschlossene, fehlgeschlagene oder abgebrochene Aufträge. Weitere Informationen finden Sie unter [Beispiele: Abschlussberichte zu S3-Batchvorgängen](#).

Themen

- [Batch-Vorgangsauftrag-Anforderungselemente](#)
- [Angaben eines Manifests](#)

Batch-Vorgangsauftrag-Anforderungselemente

Um einen S3-Batch-Vorgangsauftrag zu erstellen, müssen Sie die folgenden Informationen angeben:

Operation

Geben Sie die Operation an, die S3-Batch-Vorgänge für die Objekte im Manifest ausführen soll. Jeder Operationstyp akzeptiert Parameter, die für diesen Vorgang spezifisch sind. Mit Batch-Operationen können Sie einen Vorgang in großen Mengen ausführen, mit den gleichen Ergebnissen, als ob Sie diesen Vorgang one-by-one für jedes Objekt ausgeführt hätten.

Manifest

Das Manifest ist eine Liste aller Objekte, für die S3 Batch Operations die festgelegte Aktion ausführen soll. Sie können die folgenden Verfahren verwenden, um ein Manifest für einen Batch-Operations-Auftrag anzugeben:

- Erstellen Sie manuell Ihre eigene benutzerdefinierte, CSV formatierte Objektliste.
- Wählen Sie einen vorhandenen CSV [Amazon S3 Inventory](#) -formatierten Bericht aus.
- Weisen Sie Batch Operations an, automatisch ein Manifest auf der Grundlage von Objektfiterkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3 S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mit dem AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI.

Note

- Unabhängig davon, wie Sie Ihr Manifest angeben, muss die Liste selbst in einem Allzweck-Bucket gespeichert werden. Batch Operations kann keine vorhandenen Manifeste aus Verzeichnis-Buckets importieren (oder generierte Manifeste in Verzeichnis-Buckets speichern). Im Manifest beschriebene Objekte können jedoch in Verzeichnis-Buckets gespeichert werden. Weitere Informationen finden Sie unter [Verzeichnis-Buckets](#).
- Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, weist die Angabe der Version IDs für die Objekte Batch Operations an, den Vorgang für eine bestimmte Version auszuführen. Wenn keine Version IDs angegeben ist, führt Batch Operations den Vorgang mit der neuesten Version der Objekte aus. Wenn Ihr Manifest ein Versions-ID-Feld enthält, müssen Sie eine Versions-ID für alle Objekte im Manifest angeben.

Weitere Informationen finden Sie unter [Angaben eines Manifests](#).

Priorität

Mit Auftragsprioritäten können Sie angeben, welche relative Priorität dieser Auftrag gegenüber den anderen in Ihrem Konto ausgeführten Aufträgen besitzt. Eine höhere Nummer bedeutet eine höhere Priorität.

Auftragsprioritäten sind nur in Beziehung zu den für andere Aufträge in demselben Konto und in der derselben Region festgelegten Prioritäten bedeutsam. Sie können wählen, welches Nummerierungssystem für Sie funktioniert. Beispielsweise können Sie allen Restore (RestoreObject) -Aufträgen die Priorität 1, allen Copy (CopyObject) -Aufträgen die Priorität 2 und allen Replace-Zugriffskontrolllisten (ACLs) (**PutObjectAcl**) -Aufträgen die Priorität 3 zuweisen.

S3 Batch Operations priorisiert Aufträge gemäß den Prioritätszahlen, garantiert aber keine strikte Sortierung. Daher sind Auftrags-Prioritäten nicht dazu geeignet, sicherzustellen, dass ein Auftrag vor einem anderen Auftrag gestartet oder beendet wird. Wenn Sie eine strikte Sortierung gewährleisten möchten, müssen Sie den Abschluss eines Auftrags abwarten, bevor Sie den nächsten starten.

RoleArn

Geben Sie ein AWS Identity and Access Management (IAM) Rolle, um den Job auszuführen. Die IAM Rolle, die Sie verwenden, muss über ausreichende Berechtigungen verfügen, um den im Job angegebenen Vorgang auszuführen. Um beispielsweise einen CopyObject Job auszuführen, muss die IAM Rolle über die `s3:GetObject` Berechtigung für den Quell-Bucket und die `s3:PutObject` Berechtigung für den Ziel-Bucket verfügen. Außerdem braucht die Rolle die Berechtigungen, das Manifest zu lesen und den Auftragsabschlussbericht zu schreiben.

Weitere Informationen zu IAM [IAMRollen](#) finden Sie im IAMBenutzerhandbuch unter Rollen.

Weitere Informationen zu Amazon-S3-Berechtigungen finden Sie unter [Politische Maßnahmen für Amazon S3](#).


Note

Batch-Operations-Aufträge, die Aktionen für Verzeichnis-Buckets ausführen, erfordern bestimmte Berechtigungen. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

Bericht


Geben Sie an, ob S3-Batch-Vorgänge einen Abschlussbericht erstellen soll. Wenn Sie einen Auftragsabschlussbericht anfordern, müssen Sie auch die Parameter für den Bericht in diesem Element angeben. Die notwendigen Informationen umfassen:

- Den Bucket, in dem der Bericht gespeichert werden soll

 Note

Der Bericht muss in einem Allzweck-Bucket gespeichert werden. Batch Operations kann Berichte nicht in Verzeichnis-Buckets speichern. Weitere Informationen finden Sie unter [Verzeichnis-Buckets](#).

- Das Format des Berichts
- Ob der Bericht die Details aller oder nur fehlgeschlagener Aufgaben enthält
- Eine optionale Präfix-Zeichenfolge

 Note

Abschlussberichte werden immer mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt.

Markierungen (optional)

Sie können den Zugriff auf Ihre S3-Batch-Vorgänge-Aufträge kennzeichnen und steuern, indem Sie Markierungen hinzufügen. Sie können Tags verwenden, um zu identifizieren, wer für einen Batch-Operations-Auftrag verantwortlich ist, oder um zu steuern, wie Benutzer mit einzelnen Batch-Operations-Aufträgen interagieren. Das Vorhandensein von Aufgabe-Markierungen kann einem Benutzer die Möglichkeit gewähren oder einschränken, eine Aufgabe abzubrechen, eine Aufgabe im Bestätigungsstatus zu aktivieren oder die Prioritätsstufe einer Aufgabe zu ändern. Beispielsweise können Sie einem Benutzer die Berechtigung zum Aufrufen des `CreateJob`-Vorgangs erteilen, vorausgesetzt, der Auftrag wird mit dem Tag "Department=Finance" erstellt.

Sie können Aufträge mit bereits zugeordneten Markierungen erstellen und Sie können bereits erstellten Aufträgen nachträglich Markierungen hinzufügen.

Weitere Informationen finden Sie unter [the section called "Verwenden von Markierungen"](#).

Description (optional)

Um Ihren Auftrag zu verfolgen und zu überwachen, können Sie auch eine Beschreibung von bis zu 256 Zeichen angeben. Amazon S3 enthält diese Beschreibung immer dann, wenn

Informationen über einen Auftrag zurückgegeben oder Auftragsdetails auf der Amazon-S3-Konsole angezeigt werden. So können Sie Aufträge ganz einfach nach den ihnen zugewiesenen Beschreibungen sortieren und filtern. Beschreibungen müssen nicht eindeutig sein, sodass Sie Beschreibungen auch als Kategorien (z. B. "Wöchentliche Aufträge zum Kopieren von Protokollen") verwenden können, um Gruppen ähnlicher Aufträge nachzuverfolgen.

Angeben eines Manifests

Ein Manifest ist ein Amazon-S3-Objekt, das Objektschlüssel enthält, die Amazon S3 bearbeiten soll. Sie können ein Manifest mit einer der folgenden Methoden bereitstellen:

- Erstellen Sie eine neue Manifestdatei manuell.
- Verwenden Sie ein vorhandenes Manifest.
- Weisen Sie Batch Operations an, automatisch ein Manifest auf der Grundlage von Objektfiterkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3 S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mit dem AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI.

Note

Unabhängig davon, wie Sie Ihr Manifest angeben, muss die Liste selbst in einem Allzweck-Bucket gespeichert werden. Batch Operations kann keine vorhandenen Manifeste aus Verzeichnis-Buckets importieren (oder generierte Manifeste in Verzeichnis-Buckets speichern). Im Manifest beschriebene Objekte können jedoch in Verzeichnis-Buckets gespeichert werden. Weitere Informationen finden Sie unter [Verzeichnis-Buckets](#).

Erstellen einer Manifestdatei

Um eine Manifestdatei manuell zu erstellen, geben Sie den Manifest-Objektschlüssel ETag (Entity-Tag) und die optionale Versions-ID in einer CSV -formatierten Liste an. Der Inhalt des Manifests muss -codiert sein. URL

Standardmäßig verwendet Amazon S3 automatisch serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), um ein Manifest zu verschlüsseln, das in einen Amazon S3 S3-Bucket hochgeladen wird. Manifeste, die serverseitige Verschlüsselung mit vom Kunden

bereitgestellten Schlüsseln (SSE-C) verwenden, werden nicht unterstützt. Manifeste, die serverseitige Verschlüsselung verwenden mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) werden nur unterstützt, wenn Sie Inventarberichte im CSV -Format verwenden. Verwenden Sie ein manuell erstelltes Manifest mit AWS KMS wird nicht unterstützt.

Ihr Manifest muss den Bucket-Namen, den Objektschlüssel und (optional) die Objektversion enthalten. Sämtliche anderen Felder in dem Manifest werden nicht von S3-Batch-Vorgänge verwendet.

Note

Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, weist die Angabe der Version IDs für die Objekte Batch Operations an, den Vorgang für eine bestimmte Version auszuführen. Wenn keine Version IDs angegeben ist, führt Batch Operations den Vorgang mit der neuesten Version der Objekte aus. Wenn Ihr Manifest ein Versions-ID-Feld enthält, müssen Sie eine Versions-ID für alle Objekte im Manifest angeben.

Im Folgenden finden Sie ein Beispiel für ein Manifest im CSV Format ohne VersionIDs.

```
Examplebucket,objectkey1
Examplebucket,objectkey2
Examplebucket,objectkey3
Examplebucket,photos/jpgs/objectkey4
Examplebucket,photos/jpgs/newjersey/objectkey5
Examplebucket,object%20key%20with%20spaces
```

Das Folgende ist ein Beispiel für ein Manifest in einem CSV Format, das die Version enthältIDs.

```
Examplebucket,objectkey1,PZ9ibn9D51P6p298B7S9_ceqx1n5EJ0p
Examplebucket,objectkey2,YY_ouuAJByNW1LRBfFMfxMge7XQWxMBF
Examplebucket,objectkey3,jbo9_jhdPEyB4Rim0xWS0kU0EoNrU_oI
Examplebucket,photos/jpgs/objectkey4,6Eq1ikJJxLTsHsnbZbSRffn24_eh5Ny4
Examplebucket,photos/jpgs/newjersey/objectkey5,imHf3FAiRsvBW_EHB8G0u.NHunH01gVs
Examplebucket,object%20key%20with%20spaces,9HkPvDaZY5MVbMhn6TMn1YTb5A1rQao3w
```

Angeben einer vorhandenen Manifestdatei

Sie können ein Manifest in einer Auftragserstellungsanforderung in einem der beiden folgenden Formate erstellen:

- **Amazon S3 S3-Inventarbericht** — Muss ein Amazon S3 S3-Inventarbericht CSV im -Format sein. Sie müssen die `manifest.json`-Datei angeben, die mit dem Bestandsbericht verknüpft ist. Weitere Informationen zu Bestandsberichten finden Sie unter [Amazon S3 Inventory](#). Wenn der Inventarbericht eine Version enthält IDs, arbeitet S3 Batch Operations mit den spezifischen Objektversionen.

Note

- S3 Batch Operations unterstützt CSV Inventarberichte, die mit SSE - verschlüsselt sind KMS.
- Wenn Sie ein mit SSE — verschlüsseltes Inventarberichtsmanifest einreichen KMS, muss Ihre IAM Richtlinie die Berechtigungen `"kms:Decrypt"` und `"kms:GenerateDataKey"` für das `manifest.json` Objekt und alle zugehörigen CSV Datendateien enthalten.

- **CSVDatei** — Jede Zeile in der Datei muss den Bucket-Namen, den Objektschlüssel und optional die Objektversion enthalten. Objektschlüssel müssen URL -kodiert sein, wie in den folgenden Beispielen gezeigt. Das Manifest muss entweder die Version IDs für alle Objekte enthalten oder die Version für alle Objekte weglassen. IDs Weitere Hinweise zum CSV Manifestformat finden Sie unter [JobManifestSpec](#) in der Amazon Simple Storage Service API Reference.

Note

- S3 Batch Operations unterstützt keine CSV Manifestdateien, die mit SSE - verschlüsselt sind KMS.

⚠ Important

Wenn Sie ein manuell erstelltes Manifest und einen versionierten Bucket verwenden, empfehlen wir Ihnen, die Version IDs für die Objekte anzugeben. Wenn Sie einen Auftrag erstellen, analysiert S3-Batch-Vorgänge das gesamte Manifest, bevor Sie den Auftrag ausführen. Es wird jedoch kein „Snapshot“ des Bucket-Zustands aufgenommen. Da Manifeste Milliarden von Objekten enthalten können, kann die Ausführung von Aufträgen längere Zeit dauern, was sich darauf auswirken kann, für welche Version eines Objekts der Auftrag durchgeführt wird. Angenommen, Sie überschreiben ein Objekt mit einer neuen Version, während ein Auftrag ausgeführt wird, und Sie haben keine Versions-ID für dieses

Objekt angegeben. In diesem Fall führt Amazon S3 den Vorgang auf der neuesten Version des Objekts durch, und nicht auf der Version, die galt, als Sie den Auftrag erstellten. Die einzige Möglichkeit, dieses Verhalten zu vermeiden, besteht darin, die Version IDs für die Objekte anzugeben, die im Manifest aufgeführt sind.

Automatisches Generieren eines Manifests

Sie können Amazon S3 Batch Operations anweisen, automatisch ein Manifest auf der Grundlage von Objektfilterkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3 S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mit dem AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI. Weitere Informationen zur Batch-Replikation finden Sie unter [Replizieren bestehender Objekte mit S3-Batch-Replikation](#).

Um ein Manifest automatisch zu generieren, geben Sie im Rahmen Ihrer Anforderung zur Auftragserstellung die folgenden Elemente an:

- Informationen über den Bucket, der Ihre Quellobjekte enthält, einschließlich des Bucket-Besitzers und des Amazon-Ressourcennamens (ARN)
- Informationen zur Manifestausgabe, einschließlich eines Flags zum Erstellen einer Manifestdatei, des Besitzers des Ausgabe-BucketsARN, des Präfixes, des Dateiformats und des Verschlüsselungstyps
- Optionale Kriterien zum Filtern von Objekten nach Erstellungsdatum, Schlüsselname, Größe und Speicherklasse

Objektfilterkriterien

Um die Liste der Objekte zu filtern, die in ein automatisch generiertes Manifest aufgenommen werden sollen, können Sie die folgenden Kriterien angeben. Weitere Informationen finden Sie unter [JobManifestGeneratorFilter](#) in der Amazon S3 API S3-Referenz.

CreatedAfter

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die nach diesem Zeitpunkt erstellt wurden.

CreatedBefore

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die vor diesem Zeitpunkt erstellt wurden.

EligibleForReplication

Falls angegeben, enthält das generierte Manifest nur Objekte, wenn sie gemäß der Replikationskonfiguration im Quell-Bucket für die Replikation in Frage kommen.

KeyNameConstraint

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, deren Objektschlüssel den für `MatchAnySubstringMatchAnyPrefix`, und angegebenen Zeichenketteneinschränkungen entsprechen `MatchAnySuffix`.

MatchAnySubstring— Falls angegeben, enthält das generierte Manifest Objekte, wenn die angegebene Zeichenfolge an einer beliebigen Stelle in der Objektschlüsselzeichenfolge vorkommt.

MatchAnyPrefix— Falls angegeben, enthält das generierte Manifest Objekte, wenn die angegebene Zeichenfolge am Anfang der Objektschlüsselzeichenfolge steht.

MatchAnySuffix— Falls angegeben, enthält das generierte Manifest Objekte, wenn die angegebene Zeichenfolge am Ende der Objektschlüsselzeichenfolge steht.

MatchAnyStorageClass

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die mit der angegebenen Speicherklasse gespeichert sind.

ObjectReplicationStatuses

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die einen der angegebenen Replikationsstatus haben.

ObjectSizeGreaterThanBytes

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, deren Dateigröße die angegebene Byteanzahl überschreitet.

ObjectSizeLessThanBytes

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, deren Dateigröße die angegebene Byteanzahl unterschreitet.

Note

Die meisten Aufträge mit automatisch generierten Manifesten können nicht geklont werden. Stapelreplikationsaufträge können geklont werden, es sei denn, sie verwenden die Manifest-Filterkriterien `KeyNameConstraint`, `MatchAnyStorageClass`, `ObjectSizeGreaterThanBytes` oder `ObjectSizeLessThanBytes`.

Die Syntax für die Angabe von Manifestkriterien hängt von der Methode ab, mit der Sie Ihren Job erstellen. Beispiele finden Sie unter [Erstellen eines-Auftrags](#).

Erstellen eines-Auftrags

Sie können S3-Batch Operations-Jobs mithilfe der Amazon S3 S3-Konsole erstellen. AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI.

Weitere Informationen zum Erstellen einer Auftragsanfrage finden Sie unter [Batch-Vorgangsauftrag-Anforderungselemente](#).

Voraussetzungen

Bevor Sie einen Batch-Operations-Auftrag erstellen, bestätigen Sie, dass Sie die entsprechenden Berechtigungen konfiguriert haben. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen für Amazon-S3-BatchVorgänge](#).

Verwenden der S3-Konsole


So erstellen Sie einen Batch-Auftrag:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen des aktuell angezeigten AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie Ihren Job erstellen möchten.

Note

Für Kopiervorgänge müssen Sie den Job in derselben Region wie den Ziel-Bucket erstellen. Für alle anderen Operationen müssen Sie den Job in derselben Region wie die Objekte im Manifest erstellen.

3. Wählen Sie im linken Navigationsbereich der Amazon S3 S3-Konsole Batch Operations aus.
4. Wählen Sie Job erstellen aus.
5. Aufrufen des AWS-Regionwo Sie Ihren Job erstellen möchten.
6. Wählen Sie unter Manifest format (Manifestformat) das zu verwendende Manifestobjekt aus.
 - Wenn Sie den S3-Inventarbericht wählen, geben Sie den Pfad zum manifest.json-Objekt ein, das Amazon S3 als Teil des CSV -formatierten Inventarberichts generiert hat, und optional die Versions-ID für das Manifest-Objekt, wenn Sie eine andere Version als die neueste verwenden möchten.
 - Wenn Sie möchten CSV, geben Sie den Pfad zu einem Manifest-Objekt im -Format einCSV. Das Manifestobjekt muss das in der Konsole beschriebene Format befolgen. Sie können zudem die Versions-ID des Manifestobjekts angeben, wenn Sie eine andere Version als die aktuelle verwenden möchten.

 Note

Die Amazon-S3-Konsole unterstützt die automatische Manifestgenerierung nur für Stapelreplikationsaufträge. Wenn Sie möchten, dass Amazon S3 für alle anderen Auftragsstypen automatisch ein Manifest generiert, das auf den von Ihnen angegebenen Filterkriterien basiert, müssen Sie Ihren Job mit dem AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI.

7. Wählen Sie Weiter.
8. Wählen Sie unter Operation (Vorgang) den Vorgang aus, den Sie für alle Objekte in dem Manifest ausführen möchten. Geben Sie die Informationen für die ausgewählte Operation ein und wählen Sie anschließend Next (Weiter) aus.
9. Geben Sie die Informationen für Configure additional options (Zusätzliche Optionen konfigurieren) ein und wählen Sie anschließend Next (Weiter) aus.
10. Überprüfen Sie die Einstellungen unter Review (Überprüfen). Wenn Sie Änderungen vornehmen müssen, wählen Sie Previous. Wählen Sie andernfalls Create Job (Auftrag erstellen).

Verwendung der AWS CLI

Specify manifest

Das folgende Beispiel zeigt, wie Sie einen S3-Batch-Operations–S3PutObjectTaggingAuftrag für Objekte erstellen, die in einer vorhandenen Manifestdatei aufgelistet sind.

So erstellen Sie den Batch-Vorgangsauftrag **S3PutObjectTagging**


1. Verwenden Sie die folgenden Befehle, um ein AWS Identity and Access Management (IAM) Rolle und erstellen Sie dann eine IAM Richtlinie, um die entsprechenden Berechtigungen zuzuweisen. Die folgende Rolle und die folgende Richtlinie gewähren Amazon S3 die Berechtigung zum Hinzufügen von Objekt-Tags, was Sie benötigen, wenn Sie in einem nachfolgenden Schritt den Auftrag erstellen.
 - a. Verwenden Sie den folgenden Beispielbefehl, um eine IAM Rolle zu erstellen, die Batch Operations verwenden kann. Ersetzen Sie *S3BatchJobRole* durch den Namen, den Sie der Rolle geben möchten, um diesen Beispielbefehl zu verwenden.

```
aws iam create-role \  
  --role-name S3BatchJobRole \  
  --assume-role-policy-document '{  
    "Version":"2012-10-17",  
    "Statement":[  
      {  
        "Effect":"Allow",  
        "Principal":{  
          "Service":"batchoperations.s3.amazonaws.com"  
        },  
        "Action":"sts:AssumeRole"  
      }  
    ]  
  }'  
'
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Rolle. Sie benötigen den ARN, wenn Sie einen Job erstellen.

- b. Verwenden Sie den folgenden Beispielbefehl, um eine IAM Richtlinie mit den erforderlichen Berechtigungen zu erstellen und sie der IAM Rolle zuzuordnen, die Sie im vorherigen Schritt erstellt haben. Weitere Informationen zu den erforderlichen

Berechtigungen finden Sie unter [Erteilen von Berechtigungen für Amazon-S3-BatchVorgänge](#).

 Note

Batch-Operations-Aufträge, die Aktionen für Verzeichnis-Buckets ausführen, erfordern bestimmte Berechtigungen. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

Um diesen Beispielbefehl zu verwenden, ersetzen Sie die *user input placeholders* wie folgt:

- *S3BatchJobRole* Ersetzen Sie es durch den Namen Ihrer IAM Rolle. Stellen Sie sicher, dass dieser Name mit dem Namen übereinstimmt, den Sie zuvor verwendet haben.
- *PutObjectTaggingBatchJobPolicy* Ersetzen Sie es durch den Namen, den Sie Ihrer IAM Richtlinie geben möchten.
- Ersetzen Sie *amzn-s3-demo-destination-bucket* durch den Namen des Buckets, der die Objekte enthält, auf die Sie Tags anwenden möchten.
- Ersetzen Sie *DOC-EXAMPLE-MANIFEST-BUCKET* durch den Namen des Buckets, der das Manifest enthält.
- Ersetzen Sie *DOC-EXAMPLE-REPORT-BUCKET* durch den Namen des Buckets, an den der Abschlussbericht übermittelt werden soll.

```
aws iam put-role-policy \  
  --role-name S3BatchJobRole \  
  --policy-name PutObjectTaggingBatchJobPolicy \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Action": [  
          "s3:PutObjectTagging",  
          "s3:PutObjectVersionTagging"  
        ],  
      }  
    ],  
  }'
```

```

    "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion",
      "s3:GetBucketLocation"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:GetBucketLocation"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*"
    ]
  }
]
}'

```

2. Verwenden Sie den folgenden Beispielbefehl, um einen S3PutObjectTagging-Auftrag zu erstellen.

Die `manifest.csv`-Datei stellt eine Liste mit Bucket- und Objektschlüsselwerten bereit. Der Auftrag wendet die angegebenen Tags auf Objekte an, die im Manifest aufgelistet sind. Das ETag ist das ETag `manifest.csv` Objekt, das Sie von der Amazon S3 S3-Konsole abrufen können. Diese Anforderung enthält den `no-confirmation-required`-Parameter, damit Sie den Job ausführen können, ohne ihn mit dem `update-job-status`-Befehl bestätigen zu müssen. Weitere Informationen finden Sie unter [create-job](#) in der AWS CLI Befehlsreferenz.

Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen. Ersetzen *IAM-Rolle* mit ARN der IAM Rolle, die Sie zuvor erstellt haben.

```
aws s3control create-job \
  --region us-west-2 \
  --account-id acct-id \
  --operation '{"S3PutObjectTagging": { "TagSet": [{"Key": "keyOne",
"Value": "ValueOne"}] }}' \
  --manifest '{"Spec":{"Format": "S3BatchOperations_CSV_20180820", "Fields":
["Bucket", "Key"]}, "Location":
{"ObjectArn": "arn:aws:s3:::my_manifests/
manifest.csv", "ETag": "60e460c9d1046e73f7dde5043ac3ae85"}}' \
  --report '{"Bucket": "arn:aws:s3:::DOC-EXAMPLE-REPORT-
BUCKET", "Prefix": "final-reports",
"Format": "Report_CSV_20180820", "Enabled": true, "ReportScope": "AllTasks"}' \
  --priority 42 \
  --role-arn IAM-role \
  --client-request-token $(uuidgen) \
  --description "job description" \
  --no-confirmation-required
```

Als Antwort gibt Amazon S3 eine Auftrags-ID zurück (zum Beispiel, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Sie benötigen die Auftrags-ID, um den Auftrag zu identifizieren, zu überwachen und zu ändern.

Generate manifest

Das folgende Beispiel zeigt, wie Sie einen `S3BatchOperations-S3DeleteObjectTagging`-Auftrag erstellen, der automatisch ein Manifest auf der Grundlage Ihrer Objektkriterien generiert. Zu diesen Kriterien gehören das Erstellungsdatum, der Schlüsselname, die Größe, die Speicherklasse und die Tags.

So erstellen Sie den Batch-Vorgangsauftrag **S3DeleteObjectTagging**

1. Verwenden Sie die folgenden Befehle, um eine zu erstellen AWS Identity and Access Management (IAM) Rolle und erstellen Sie dann eine IAM Richtlinie zum Zuweisen von Berechtigungen. Die folgende Rolle und die folgende Richtlinie gewähren Amazon S3 die Berechtigung zum Löschen von Objekt-Tags, was Sie benötigen, wenn Sie in einem nachfolgenden Schritt den Auftrag erstellen.


a.

Verwenden Sie den folgenden Beispielbefehl, um eine IAM Rolle zu erstellen, die Batch Operations verwenden kann. Ersetzen Sie *S3BatchJobRole* durch den Namen, den Sie der Rolle geben möchten, um diesen Beispielbefehl zu verwenden.

```
aws iam create-role \  
  --role-name S3BatchJobRole \  
  --assume-role-policy-document '{  
    "Version":"2012-10-17",  
    "Statement":[  
      {  
        "Effect":"Allow",  
        "Principal":{"  
          "Service":"batchoperations.s3.amazonaws.com"  
        }  
      },  
      "Action":"sts:AssumeRole"  
    ]  
  }'  
'
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Rolle. Sie benötigen den ARN, wenn Sie einen Job erstellen.

- b. Verwenden Sie den folgenden Beispielbefehl, um eine IAM Richtlinie mit den erforderlichen Berechtigungen zu erstellen und sie der IAM Rolle zuzuordnen, die Sie im vorherigen Schritt erstellt haben. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [Erteilen von Berechtigungen für Amazon-S3-Batchvorgänge](#).

 Note

Batch-Operations-Aufträge, die Aktionen für Verzeichnis-Buckets ausführen, erfordern bestimmte Berechtigungen. Weitere Informationen finden Sie unter [AWS Identity and Access Management \(IAM\) für S3 Express One Zone](#).

Um diesen Beispielbefehl zu verwenden, ersetzen Sie die *user input placeholders* wie folgt:

- *S3BatchJobRole* Ersetzen Sie es durch den Namen Ihrer IAM Rolle. Stellen Sie sicher, dass dieser Name mit dem Namen übereinstimmt, den Sie zuvor verwendet haben.
- *DeleteObjectTaggingBatchJobPolicy* Ersetzen Sie es durch den Namen, den Sie Ihrer IAM Richtlinie geben möchten.
- Ersetzen Sie *amzn-s3-demo-destination-bucket* durch den Namen des Buckets, der die Objekte enthält, auf die Sie Tags anwenden möchten.
- Ersetzen Sie *DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET* durch den Namen des Buckets, in dem das Manifest gespeichert werden soll.
- Ersetzen Sie *DOC-EXAMPLE-REPORT-BUCKET* durch den Namen des Buckets, an den der Abschlussbericht übermittelt werden soll.

```
aws iam put-role-policy \  
  --role-name S3BatchJobRole \  
  --policy-name DeleteObjectTaggingBatchJobPolicy \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Action": [  
          "s3:DeleteObjectTagging",  
          "s3:DeleteObjectVersionTagging"  
        ],  
        "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"  
      },  
      {  
        "Effect": "Allow",  
        "Action": [  
          "s3:PutInventoryConfiguration"  
        ],  
        "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket"  
      },  
      {  
        "Effect": "Allow",  
        "Action": [  
          "s3:GetObject",  
          "s3:GetObjectVersion",  
          "s3:ListBucket"  
        ]  
      }  
    ]  
  }'
```

```

    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*",
      "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"
    ]
  }
]
}'

```

2. Verwenden Sie den folgenden Beispielbefehl, um den S3DeleteObjectTagging-Auftrag zu erstellen.

In diesem Beispiel geben die Werte im `--report`-Abschnitt den Bucket, das Präfix, das Format und den Umfang des Auftragsberichts an, der generiert wird. Der `--manifest-generator`-Abschnitt enthält Informationen über den Quell-Bucket, der die Objekte enthält, auf die der Auftrag wirkt, Informationen über die Manifest-Ausgabeliste, die für den Auftrag generiert wird, und Filterkriterien, um den Bereich der Objekte, die in das Manifest aufgenommen werden sollen, nach Erstellungsdatum, Namenseinschränkungen, Größe und Speicherklasse einzugrenzen. Der Befehl gibt auch die Priorität, IAM Rolle und AWS-Region.

Weitere Informationen finden Sie unter [create-job](#) in der AWS CLI Befehlsreferenz.

Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen. Ersetzen *IAM-Rolle* mit ARN der IAM Rolle, die Sie zuvor erstellt haben.

```

aws s3control create-job \
  --account-id 012345678901 \
  --operation '{

```

```

    "S3DeleteObjectTagging": {}
  }' \
  --report '{
    "Bucket": "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
    "Prefix": "reports",
    "Format": "Report_CSV_20180820",
    "Enabled": true,
    "ReportScope": "AllTasks"
  }' \
  --manifest-generator '{
    "S3JobManifestGenerator": {
      "ExpectedBucketOwner": "012345678901",
      "SourceBucket": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
      "EnableManifestOutput": true,
      "ManifestOutputLocation": {
        "ExpectedManifestBucketOwner": "012345678901",
        "Bucket": "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET",
        "ManifestPrefix": "prefix",
        "ManifestFormat": "S3InventoryReport_CSV_20211130"
      },
      "Filter": {
        "CreatedAfter": "2023-09-01",
        "CreatedBefore": "2023-10-01",
        "KeyNameConstraint": {
          "MatchAnyPrefix": [
            "prefix"
          ],
          "MatchAnySuffix": [
            "suffix"
          ]
        },
        "ObjectSizeGreaterThanBytes": 100,
        "ObjectSizeLessThanBytes": 200,
        "MatchAnyStorageClass": [
          "STANDARD",
          "STANDARD_IA"
        ]
      }
    }
  }' \
  --priority 2 \
  --role-arn IAM-role \
  --region us-east-1

```


Als Antwort gibt Amazon S3 eine Auftrags-ID zurück (zum Beispiel, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Sie benötigen diese Auftrags-ID, um den Auftrag zu identifizieren, zu überwachen oder zu ändern.

Verwendung der AWS SDK for Java

Specify manifest

Das folgende Beispiel zeigt, wie Sie einen S3-Batch-Operations-S3PutObjectTagging-Auftrag für Objekte erstellen, die in einer vorhandenen Manifestdatei aufgelistet sind. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.*;

import java.util.UUID;
import java.util.ArrayList;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateJob {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String iamRoleArn = "IAM Role ARN";
        String reportBucketName = "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET";
        String uuid = UUID.randomUUID().toString();

        ArrayList tagSet = new ArrayList<S3Tag>();
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));
    }
}
```

```
try {
    JobOperation jobOperation = new JobOperation()
        .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
            .withTagSet(tagSet)
        );

    JobManifest manifest = new JobManifest()
        .withSpec(new JobManifestSpec()
            .withFormat("S3BatchOperations_CSV_20180820")
            .withFields(new String[]{
                "Bucket", "Key"
            })
        .withLocation(new JobManifestLocation()
            .withObjectArn("arn:aws:s3:::my_manifests/manifest.csv")
            .withETag("60e460c9d1046e73f7dde5043ac3ae85"));

    JobReport jobReport = new JobReport()
        .withBucket(reportBucketName)
        .withPrefix("reports")
        .withFormat("Report_CSV_20180820")
        .withEnabled(true)
        .withReportScope("AllTasks");

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.createJob(new CreateJobRequest()
        .withAccountId(accountId)
        .withOperation(jobOperation)
        .withManifest(manifest)
        .withReport(jobReport)
        .withPriority(42)
        .withRoleArn(iamRoleArn)
        .withClientRequestToken(uuid)
        .withDescription("job description")
        .withConfirmationRequired(false)
    );

} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
```

```
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

Generate manifest

Das folgende Beispiel zeigt, wie Sie einen S3-Batch-Operations–s3PutObjectCopy-Auftrag erstellen, der automatisch ein Manifest auf der Grundlage Ihrer Objektfilterkriterien, einschließlich des Erstellungsdatums, des Schlüsselnamens und der Größe, generiert. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.CreateJobRequest;
import com.amazonaws.services.s3control.model.CreateJobResult;
import com.amazonaws.services.s3control.model.JobManifestGenerator;
import com.amazonaws.services.s3control.model.JobManifestGeneratorFilter;
import com.amazonaws.services.s3control.model.JobOperation;
import com.amazonaws.services.s3control.model.JobReport;
import com.amazonaws.services.s3control.model.KeyNameConstraint;
import com.amazonaws.services.s3control.model.S3JobManifestGenerator;
import com.amazonaws.services.s3control.model.S3ManifestOutputLocation;
import com.amazonaws.services.s3control.model.S3SetObjectTaggingOperation;
import com.amazonaws.services.s3control.model.S3Tag;

import java.time.Instant;
import java.util.Date;
import java.util.UUID;
import java.util.ArrayList;
```

```
import static com.amazonaws.regions.Regions.US_WEST_2;  
  
public class test {  
    public static void main(String[] args) {  
        String accountId = "012345678901";  
        String iamRoleArn = "arn:aws:iam::012345678901:role/ROLE";  
        String sourceBucketName = "arn:aws:s3::DOC-EXAMPLE-SOURCE-BUCKET";  
        String reportBucketName = "arn:aws:s3::DOC-EXAMPLE-REPORT-BUCKET";  
        String manifestOutputBucketName = "arn:aws:s3::DOC-EXAMPLE-MANIFEST-  
OUTPUT-BUCKET";  
        String uuid = UUID.randomUUID().toString();  
        long minimumObjectSize = 100L;  
  
        ArrayList<S3Tag> tagSet = new ArrayList<>();  
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));  
  
        ArrayList<String> prefixes = new ArrayList<>();  
        prefixes.add("s3KeyStartsWith");  
  
        try {  
            JobOperation jobOperation = new JobOperation()  
                .withS3PutObjectTagging(new S3SetObjectTaggingOperation()  
                    .withTagSet(tagSet)  
                );  
            S3ManifestOutputLocation manifestOutputLocation = new  
S3ManifestOutputLocation()  
                .withBucket(manifestOutputBucketName)  
                .withManifestPrefix("manifests")  
                .withExpectedManifestBucketOwner(accountId)  
                .withManifestFormat("S3InventoryReport_CSV_20211130");  
  
            JobManifestGeneratorFilter jobManifestGeneratorFilter = new  
JobManifestGeneratorFilter()  
                .withEligibleForReplication(true)  
                .withKeyNameConstraint(  
                    new KeyNameConstraint()  
                        .withMatchAnyPrefix(prefixes))  
                .withCreatedBefore(Date.from(Instant.now()))  
                .withObjectSizeGreaterThanBytes(minimumObjectSize);  
  
            S3JobManifestGenerator s3JobManifestGenerator = new  
S3JobManifestGenerator()  
                .withEnableManifestOutput(true)  
                .withManifestOutputLocation(manifestOutputLocation)
```

```
        .withFilter(jobManifestGeneratorFilter)
        .withSourceBucket(sourceBucketName);

    JobManifestGenerator jobManifestGenerator = new
JobManifestGenerator()
        .withS3JobManifestGenerator(s3JobManifestGenerator);

    JobReport jobReport = new JobReport()
        .withBucket(reportBucketName)
        .withPrefix("reports")
        .withFormat("Report_CSV_20180820")
        .withEnabled(true)
        .withReportScope("AllTasks");

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    CreateJobResult createJobResult = s3ControlClient.createJob(new
CreateJobRequest()
        .withAccountId(accountId)
        .withOperation(jobOperation)
        .withManifestGenerator(jobManifestGenerator)
        .withReport(jobReport)
        .withPriority(42)
        .withRoleArn(iamRoleArn)
        .withClientRequestToken(uuid)
        .withDescription("job description")
        .withConfirmationRequired(true)
    );

    System.out.println("Created job " + createJobResult.getJobId());

} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't
process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
```

```
}  
}
```

Mit dem REST API

Sie können den verwenden RESTAPI, um einen Batch Operations-Job zu erstellen. Weitere Informationen finden Sie unter [CreateJob](#) in der Amazon Simple Storage Service API Reference.

Auftragsantworten

Wenn die `CreateJob`-Anforderung erfolgreich ist, gibt Amazon S3 eine Auftrags-ID zurück. Die Auftrags-ID ist eine eindeutige Kennung, die Amazon S3 automatisch erstellt. Anhand dieser ID können Sie Ihren Batch-Vorgangsauftrag identifizieren und dessen Status überwachen.

Wenn Sie einen Job über das erstellen AWS CLI, AWS SDKsRESTAPI, oder Sie können S3 Batch Operations so einrichten, dass die Verarbeitung des Jobs automatisch beginnt. Der Auftrag wird ausgeführt, sobald er bereit ist, und muss nicht hinter Aufträgen mit höherer Priorität warten.

Wenn Sie einen Auftrag über die Amazon-S3-Konsole erstellen, müssen Sie die Auftragsdetails überprüfen und bestätigen, dass Sie ihn ausführen möchten, bevor Batch Operations mit seiner Verarbeitung beginnen kann. Wenn ein Auftrag den Status „Ausgesetzt“ mehr als 30 Tage behält, schlägt er fehl.

Von S3 Batch-Vorgänge unterstützte Vorgänge

S3-Batch-Vorgänge unterstützt verschiedene Vorgänge. In den Themen in diesem Abschnitt wird jede dieser Vorgänge beschrieben.

Kopieren von Objekten

Die Operation Copy (Kopieren) kopiert jedes im Manifest angegebene Objekt. Sie können Objekte in einen Bucket in derselben AWS-Region oder in einen Bucket in einer anderen Region kopieren. S3-Batch-Vorgänge unterstützt die meisten über Amazon S3 verfügbaren Optionen zum Kopieren von Objekten. Zu diesen Optionen gehören das Festlegen von Objekt-Metadaten, das Festlegen von Berechtigungen sowie das Ändern der Speicherklasse eines Objekts.

Mit dem Kopiervorgang können Sie vorhandene nicht verschlüsselte Objekte kopieren und die neuen verschlüsselten Objekte in denselben Bucket schreiben. Weitere Informationen finden Sie unter [Encrypting Objects with Amazon S3 Batch Operations \(Verschlüsseln von Objekten mit Amazon S3 Batch Operations\)](#).

Wenn Sie Objekte kopieren, können Sie den Prüfsummenalgorithmus zur Berechnung der Prüfsumme des Objekts ändern. Wenn Objekte keine zusätzliche berechnete Prüfsumme haben, können Sie eine hinzufügen, indem Sie den von Amazon S3 zu verwendenden Prüfsummenalgorithmus angeben. Weitere Informationen finden Sie unter [Überprüfung der Objektintegrität](#).

Weitere Informationen zum Kopieren von Objekten in Amazon S3 sowie erforderliche und optionale Parameter finden Sie unter [Objekte kopieren, verschieben und umbenennen](#) in diesem Handbuch und in [CopyObject](#) in der API-Referenz für Amazon Simple Storage Service.

Beschränkungen und Einschränkungen

- Alle Quellobjekte müssen sich in einem Bucket befinden.
- Alle Zielobjekte müssen sich in einem Bucket befinden.
- Sie benötigen Lese-Berechtigungen für den Quell-Bucket und Schreib-Berechtigungen für den Ziel-Bucket.
- Die maximale Größe eines Objekts, das kopiert werden soll, beträgt 5 GB.
- Wenn Sie versuchen, Objekte aus den Klassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive in die Speicherklasse S3 Standard zu kopieren, müssen Sie diese Objekte zuerst wiederherstellen. Weitere Informationen finden Sie unter [Wiederherstellen eines archivierten Objekts](#).
- Copy-Aufgaben müssen in der Zielregion erstellt werden. Das ist die Region, in die Sie die Objekte kopieren möchten.
- Alle Copy-Optionen werden unterstützt, mit Ausnahme der bedingten Prüfungen von EMarkierungen und der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C).
- Wenn die Buckets nicht versioniert sind, werden die Objekte mit dem gleichen Schlüsselnamen überschrieben.
- Objekte werden nicht zwingend in derselben Reihenfolge kopiert, in der sie im Manifest erscheinen. Wenn für versionierte Buckets die Beibehaltung der aktuellen/nicht aktuellen Versionsreihenfolge wichtig ist, sollten Sie zuerst alle nicht aktuellen Versionen kopieren. Kopieren Sie dann, nachdem der erste Auftrag abgeschlossen ist, die aktuellen Versionen in einen nachfolgenden Auftrag.
- Das Kopieren von Objekten zur Klasse Reduced Redundancy Storage (RRS) wird nicht unterstützt.

Kopieren von Objekten mit S3 BatchVorgänge

Sie können S3-Batch-Vorgänge verwenden, um einen PUT-Kopierauftrag zu erstellen, um Objekte innerhalb desselben Kontos oder in ein anderes Zielkonto zu kopieren. Die folgenden Abschnitte enthalten Beispiele für die Speicherung und Verwendung eines Manifests, das sich in einem anderen Konto befindet. Im ersten Abschnitt können Sie Amazon S3 Inventory verwenden, um den Bestandsbericht für das Zielkonto bereitzustellen, damit er dort zur Auftragserstellung genutzt werden kann. Alternativ können Sie ein CSV-Manifest (Manifest mit Kommas als Trennzeichen) im Quell- oder Zielkonto verwenden, wie im zweiten Beispiel dargestellt. Das dritte Beispiel zeigt, wie der Kopiervorgang verwendet wird, um die S3-Bucket-Schlüsselverschlüsselung für vorhandene Objekte zu aktivieren.

Beispiele für Kopieroperation

- [Verwenden eines im Zielkonto bereitgestellten Bestandsberichts, um Objekte über AWS-Konten hinweg zu kopieren](#)
- [Verwenden eines CSV Manifests zum Kopieren von Objekten AWS-Konten](#)
- [Verwenden von S3-Batch-Vorgänge zum Verschlüsseln von Objekten mit S3-Bucket-Schlüssel](#)

Verwenden eines im Zielkonto bereitgestellten Bestandsberichts, um Objekte über AWS-Konten hinweg zu kopieren

Verwenden Sie Amazon S3 Inventory, um einen Bestandsbericht zu erstellen, und erstellen Sie mithilfe des Berichts eine Liste von Objekten, die mit S3-Batch-Vorgänge kopiert werden sollen. Informationen zur Verwendung eines CSV-Manifests im Quell- oder Zielkonto finden Sie unter [the section called "Verwenden eines CSV Manifests zum Kopieren von Objekten AWS-Konten"](#).

Der Amazon S3 Inventory generiert Bestandslisten der Objekte in einem Bucket. Die resultierende Liste wird in einer Ausgabedatei veröffentlicht. Der Bucket, für den die Bestandsliste erstellt wird, wird als Quell-Bucket bezeichnet, der Bucket, in dem die Bestandsberichtsdatei gespeichert wird, dagegen als Ziel-Bucket.

Der Amazon-S3-Inventory-Bericht kann so konfiguriert werden, dass er an ein anderes AWS-Konto übermittelt wird. S3-Batch-Vorgänge kann den Bestandsbericht dann lesen, wenn der Auftrag im Zielkonto erstellt wird.

Weitere Informationen zu Quell- und Ziel-Buckets für Amazon S3 Inventory finden Sie unter [Quell- und Ziel-Buckets](#).

Die einfachste Möglichkeit, einen Bestand einzurichten, ist die Verwendung der AWS Management Console, Sie können aber auch REST-API, AWS Command Line Interface (AWS CLI) oder AWS-SDKs verwenden.

Die folgende Konsolenprozedur enthält die allgemeinen Schritte zum Einrichten von Berechtigungen für einen S3-Batchoperations-Auftrag. Bei diesem Verfahren kopieren Sie Objekte aus einem Quellkonto in ein Zielkonto, wobei der Bestandsbericht im Zielkonto gespeichert wird.

So richten Sie einen Amazon S3 Inventory für Quell- und Ziel-Buckets ein, die zu unterschiedlichen Konten gehören

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie einen Ziel-Bucket, in dem der Bestandsbericht gespeichert werden soll.

Legen Sie einen Ziel-Manifest-Bucket fest, in dem der Bestandsbericht gespeichert werden soll. In diesem Verfahren ist das Zielkonto das Konto, zu dem der Ziel-Manifest-Bucket sowie der Bucket gehören, in den die Objekte kopiert werden.

3. Konfigurieren Sie eine Bestandsliste, um die Objekte in einem Quell-Bucket aufzulisten und die Liste in einem Ziel-Bucket zu veröffentlichen.

Konfigurieren Sie eine Bestandsliste für einen Quell-Bucket. Wenn Sie dies tun, geben Sie den Ziel-Bucket an, in dem die Liste gespeichert werden soll. Der Bestandsbericht für den Quell-Bucket wird im Ziel-Bucket veröffentlicht. In diesem Verfahren ist das Quellkonto das Konto, zu dem der Quell-Bucket gehört.

Informationen zur Verwendung der Konsole zum Konfigurieren einer Bestandsliste oder zum Verschlüsseln einer Bestandsdatei finden Sie unter [Konfigurieren von Amazon S3 Inventory](#).

Wählen Sie CSV als Ausgabeformat.

Wenn Sie Informationen für den Ziel-Bucket eingeben, wählen Sie Buckets in another account (Buckets in einem anderen Konto). Geben Sie dann den Namen des Ziel-Manifest-Buckets ein. Optional können Sie die Konto-ID des Zielkontos eingeben.

Nachdem die Bestands-Konfiguration gespeichert wurde, zeigt die Konsole eine Meldung wie die folgende an:

Amazon S3 could not create a bucket policy on the destination bucket. Ask the destination bucket owner to add the following bucket policy to allow Amazon S3 to place data in that

bucket. (Amazon S3 konnte keine Bucket-Richtlinie für den Ziel-Bucket erstellen. Bitten Sie den Eigentümer des Ziel-Buckets, die folgende Bucket-Richtlinie hinzuzufügen, damit Amazon S3 Daten in diesen Bucket einfügen kann.

Die Konsole zeigt dann eine Bucket-Richtlinie an, die Sie für den Ziel-Bucket verwenden können.

4. Kopieren Sie die in der Konsole angezeigte Ziel-Bucket-Richtlinie.
5. Fügen Sie die kopierte Bucket-Richtlinie im Zielkonto dem Ziel-Manifest-Bucket hinzu, in dem der Bestandsbericht gespeichert wird.
6. Erstellen Sie eine Rolle im Zielkonto, die auf der S3-BatchVorgängevertrauensrichtlinie basiert. Weitere Informationen zur Vertrauensrichtlinie finden Sie unter [Vertrauensrichtlinie](#).

Weitere Informationen zum Erstellen einer Rolle finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Geben Sie einen Namen für die Rolle ein (die Beispielrolle hat den Namen BatchOperationsDestinationRoleCOPY). Wählen Sie den Service S3 und dann den Anwendungsfall S3 Bucket Batch Operations (S3-Bucket-Batch-Vorgänge) aus, mit dem die Vertrauensrichtlinie der Rolle zugewiesen wird.

Wählen Sie dann Create policy (Richtlinie erstellen) aus, um der Rolle die folgende Richtlinie anzufügen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
      "arn:aws:s3:::ObjectDestinationBucket/*",
      "arn:aws:s3:::ObjectSourceBucket/*",
      "arn:aws:s3:::ObjectDestinationManifestBucket/*"
    ]
  }
]
}

```

Die Rolle verwendet die Richtlinie, um die Berechtigung `batchoperations.s3.amazonaws.com` zum Lesen des Manifests im Ziel-Bucket zu gewähren. Zudem werden Berechtigungen für GET-Objekte, Zugriffskontrolllisten (ACLs), Markierungen und Versionen im Quell-Bucket der Objekte gewährt. Und es werden Berechtigungen für PUT-Objekte, ACLs, Markierungen sowie Versionen im Ziel-Bucket der Objekte gewährt.

- Erstellen Sie im Quellkonto eine Bucket-Richtlinie für den Quell-Bucket, die der im vorherigen Schritt erstellten Rolle Berechtigungen für GET-Objekte, ACLs, Markierungen und Versionen im Quell-Bucket gewährt. Dieser Schritt ermöglicht es S3-BatchVorgänge, Objekte über die vertrauenswürdige Rolle aus dem Quell-Bucket abzurufen.

Das folgende Beispiel zeigt eine Bucket-Richtlinie für das Quellkonto.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceObjectCOPY",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::DestinationAccountNumber:role/
BatchOperationsDestinationRoleCOPY"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ]
    }
  ]
}

```

```
    "Resource": "arn:aws:s3:::ObjectSourceBucket/*"  
  }  
]  
}
```

8. Sobald der Bestandsbericht verfügbar ist, erstellen Sie einen S3-Batchoperations-PUT-Auftrag zum Kopieren von Objekten im Zielkonto und wählen den Bestandsbericht im Ziel-Manifest-Bucket aus. Sie benötigen den ARN der im Zielkonto erstellten Rolle.

Allgemeine Informationen zum Erstellen eines Auftrags finden Sie unter [Erstellen eines S3-Batch-Vorgangsauftrags](#).

Informationen zum Erstellen eines Auftrags mit der Konsole finden Sie unter [Erstellen eines S3-Batch-Vorgangsauftrags](#).

Verwenden eines CSV Manifests zum Kopieren von Objekten AWS-Konten

Sie können ein CSV Manifest verwenden, das im Quellkonto gespeichert ist, um Objekte AWS-Konten mit S3 Batch Operations zu kopieren. Informationen zur Verwendung eines S3-Inventarberichts als Manifest finden Sie unter [the section called "Verwenden eines Bestandsberichts, um Objekte über AWS-Konten hinweg zu kopieren"](#).

Ein Beispiel für das CSV Format für Manifestdateien finden Sie unter [the section called "Erstellen einer Manifestdatei"](#).

Das folgende Verfahren zeigt, wie Sie Berechtigungen einrichten, wenn Sie einen S3 Batch Operations-Job verwenden, um Objekte von einem Quellkonto in ein Zielkonto mit einer CSV Manifestdatei zu kopieren, die im Quellkonto gespeichert ist.

So verwenden Sie ein CSV Manifest zum Kopieren von Objekten AWS-Konten

1. Erstellen Sie eine Rolle im Zielkonto, die auf der S3-BatchVorgängevertrauensrichtlinie basiert. In diesem Verfahren ist das Zielkonto das Konto, in das die Objekte kopiert werden.

Weitere Informationen zur Vertrauensrichtlinie finden Sie unter [Vertrauensrichtlinie](#).

Weitere Informationen zum Erstellen einer Rolle finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst](#) im IAMBenutzerhandbuch.

Wenn Sie die Rolle mithilfe der Konsole erstellen, geben Sie einen Namen für die Rolle ein (die Beispielrolle verwendet den NamenBatchOperationsDestinationRoleCOPY). Wählen Sie

den Service S3 und dann den Anwendungsfall S3 Bucket Batch Operations (S3-Bucket-Batch-Vorgänge) aus, mit dem die Vertrauensrichtlinie der Rolle zugewiesen wird.

Wählen Sie dann Richtlinie erstellen, um die folgende Richtlinie an die Rolle anzuhängen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3::ObjectDestinationBucket/*",
        "arn:aws:s3::ObjectSourceBucket/*",
        "arn:aws:s3::ObjectSourceManifestBucket/*"
      ]
    }
  ]
}
```

Mit der Richtlinie gewährt die Rolle die Berechtigung `batchoperations.s3.amazonaws.com` zum Lesen des Manifests im Quell-Manifest-Bucket. Sie gewährt Berechtigungen für GET Objekte, Zugriffskontrolllisten (ACLs), Tags und Versionen im Quellobjekt-Bucket. Es gewährt auch Berechtigungen für PUT ObjekteACLs, Tags und Versionen für den Zielobjekt-Bucket.

- Erstellen Sie im Quellkonto eine Bucket-Richtlinie für den Bucket, der das Manifest enthält, um der Rolle, die Sie im vorherigen Schritt erstellt haben, Berechtigungen für GET Objekte und Versionen im Quellmanifest-Bucket zu gewähren.

In diesem Schritt kann S3 Batch Operations das Manifest mithilfe der vertrauenswürdigen Rolle lesen. Weisen Sie die Bucket-Richtlinie dem Bucket zu, in dem sich das Manifest befindet.

Im Folgenden finden Sie ein Beispiel für die Bucket-Richtlinie, die auf den Quell-Manifest-Bucket angewendet werden soll:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceManifestRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws: iam::DestinationAccountNumber:user/ConsoleUserCreatingJob",
          "arn:aws: iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws: s3::ObjectSourceManifestBucket/*"
    }
  ]
}
```

Diese Richtlinie gewährt außerdem die erforderlichen Berechtigungen, damit ein Konsolenbenutzer, der einen Auftrag im Zielkonto erstellt, diese Berechtigungen über dieselbe Bucket-Richtlinie auch für den Ziel-Manifest-Bucket erhält.

- Erstellen Sie im Quellkonto eine Bucket-Richtlinie für den Quell-Bucket, die der von Ihnen erstellten Rolle Berechtigungen für GET ObjekteACLs, Tags und Versionen im Quellobjekt-Bucket gewährt. S3-Batch-Vorgänge kann dann Objekte über die vertrauenswürdige Rolle aus dem Quell-Bucket abrufen.

Im Folgenden finden Sie ein Beispiel für die Bucket-Richtlinie für den Bucket, der die Quellobjekte enthält:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowBatchOperationsSourceObjectCOPY",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:
iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
    },
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion",
      "s3:GetObjectAcl",
      "s3:GetObjectTagging",
      "s3:GetObjectVersionAcl",
      "s3:GetObjectVersionTagging"
    ],
    "Resource": "arn:aws: s3::ObjectSourceBucket/*"
  }
]
```

4. Erstellen Sie einen S3-Batchoperations-Auftrag im Zielkonto. Sie benötigen den Amazon-Ressourcennamen (ARN) für die Rolle, die Sie im Zielkonto erstellt haben. Weitere Informationen zum Erstellen eines Auftrags finden Sie unter [Erstellen eines S3-Batch-Vorgangsauftrags](#).

Verwenden von S3-Batch-Vorgänge zum Verschlüsseln von Objekten mit S3-Bucket-Schlüssel

In diesem Abschnitt verwenden Sie den Copy-Vorgang für Amazon S3-BatchVorgänge, um die Verschlüsselung von S3-Bucket-Schlüsseln für vorhandene Objekte zu identifizieren und zu aktivieren. Weitere Informationen zu S3-Bucket-Schlüsseln finden Sie unter [Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys](#) und [Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet](#).

In diesem Beispiel werden folgende Themen behandelt:

Themen

- [Voraussetzungen](#)
- [Schritt 1: Abrufen der Liste von Objekten mithilfe von Amazon S3 Inventory](#)
- [Schritt 2: Filtern Sie Ihre Objektliste mit S3 Select](#)

- [Schritt 3: Einrichten und Ausführen des Auftrags für S3-BatchVorgänge](#)
- [Übersicht](#)

Voraussetzungen

Zum Ausführen der Schritte in diesem Verfahren benötigen Sie ein AWS-Konto und mindestens einen S3-Bucket, um Ihre Arbeitsdateien und verschlüsselten Ergebnisse zu speichern. Möglicherweise finden Sie auch vieles aus der vorhandenen Dokumentation zu S3 Batch Operations nützlich, einschließlich der folgenden Themen:

- [Grundlagen von S3-BatchVorgänge](#)
- [Erstellen eines S3-Batch-Vorgangsauftrags](#)
- [Von S3 Batch-Vorgänge unterstützte Vorgänge](#)
- [Verwalten von S3-Batch-Vorgangsaufträgen](#)

Schritt 1: Abrufen der Liste von Objekten mithilfe von Amazon S3 Inventory

Geben Sie zunächst den S3-Bucket an, der die zu verschlüsselnden Objekte enthält, und rufen Sie eine Liste des Inhalts ab. Ein Amazon S3-Bestandsbericht ist die bequemste und kostengünstigste Methode, dies zu tun. Der Bericht enthält die Liste der Objekte in einem Bucket sowie die zugehörigen Metadaten. Der Quell-Bucket bezieht sich auf den im Bestand erfassten Bucket und der Ziel-Bucket bezieht sich auf den Bucket, in dem die Bestandsberichts-Datei gespeichert wird. Weitere Informationen zu Quell- und Ziel-Buckets für Amazon S3 Inventory finden Sie unter [Amazon S3 Inventory](#).

Die einfachste Möglichkeit zum Einrichten eines Bestands ist mit der AWS Management Console. Sie können aber auch die REST-API, AWS Command Line Interface (AWS CLI), oder AWS-SDKs verwenden. Melden Sie sich bei der Konsole an und öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>. Wenn Fehler mit Berechtigung verweigert auftreten, fügen Sie eine Bucket-Richtlinie zu Ihrem Ziel-Bucket hinzu. Weitere Informationen finden Sie unter [Gewähren von Berechtigungen für S3 Inventory und S3 Analytics](#).

Abrufen der Liste von Objekten mithilfe von S3 Inventory

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich die Option Buckets und wählen Sie einen Bucket aus, der Objekte enthält, die zu verschlüsseln sind.

3. Navigieren Sie im Tab Verwaltung zu dem Abschnitt Inventory configurations (Bestands-Konfigurationen), und wählen Sie Create inventory configuration (Bestands-Konfiguration erstellen).
4. Geben Sie Ihrem neuen Bestand einen Namen, geben Sie den Namen des Ziel-S3-Buckets ein und erstellen Sie optional ein Zielpräfix für Amazon S3, um Objekte in diesem Bucket zuzuweisen.
5. Für das Ausgabeformat wählen Sie CSV.
6. (Optional) Wählen Sie im Abschnitt Zusätzliche Felder – optional die Option Verschlüsselung und alle anderen für Sie interessanten Listenfelder aus. Legen Sie die Häufigkeit für die Berichterstattung auf Daily (Täglich), damit der erste Bericht früher an Ihren Bucket geliefert wird.
7. Wählen Sie Save (Speichern) aus, um die Konfiguration zu speichern.

Amazon S3 benötigt bis zu 48 Stunden, den ersten Bericht bereitzustellen. Kehren Sie daher zurück, wenn der erste Bericht eingeht. Nachdem Sie Ihren ersten Bericht erhalten haben, fahren Sie mit dem nächsten Abschnitt fort, um den Inhalt Ihres S3-Bestandsberichts zu filtern. Wenn Sie keine Bestandsberichte für diesen Bucket mehr erhalten möchten, löschen Sie Ihre S3-Bestands-Konfiguration. Andernfalls liefert S3 Berichte nach einem täglichen oder wöchentlichen Zeitplan.

Eine Bestandsliste ist nicht eine einzelne point-in-time Ansicht aller Objekte. Bestandslisten sind fortlaufende Snapshots von Bucket-Elementen, die letztendlich konsistent sind (d. h. die Liste enthält möglicherweise keine vor kurzem hinzugefügten oder gelöschten Objekte). Die Kombination von S3 Inventory und S3-Batch-Vorgänge funktioniert am besten, wenn Sie mit statischen Objekten oder mit einem Objektsatz arbeiten, den Sie vor zwei oder mehr Tagen erstellt haben. Um mit neueren Daten zu arbeiten, verwenden Sie den API-Vorgang [ListObjectsV2](#) (GET Bucket), um Ihre Objektliste manuell zu erstellen. Wiederholen Sie den Vorgang bei Bedarf für die nächsten Tage oder bis Ihr Bestandsbericht den gewünschten Status für alle Schlüssel anzeigt.

Schritt 2: Filtern Sie Ihre Objektliste mit S3 Select

Nachdem Sie Ihren S3-Bestandsbericht erhalten haben, können Sie den Inhalt des Berichts filtern, um nur die Objekte aufzulisten, die nicht mit S3-Bucket-Schlüsseln verschlüsselt sind. Wenn Sie möchten, dass alle Objekte Ihres Buckets mit S3-Bucket-Schlüsseln verschlüsselt werden, können Sie diesen Schritt ignorieren. Durch das Filtern Ihres S3-Bestandsberichts in diesem Stadium sparen Sie jedoch Zeit und Kosten für das erneute Verschlüsseln von Objekten, die Sie zuvor verschlüsselt haben.

In den folgenden Schritten wird das Filtern mit [Amazon S3 Select](#) gezeigt. Sie können aber auch [Amazon Athena](#) verwenden. Um zu entscheiden, welches Tool verwendet werden soll, schauen Sie sich die `manifest.json`-Datei Ihres S3-Bestandsberichts an. Diese Datei listet die Anzahl der Datendateien auf, die diesem Bericht zugeordnet sind. Wenn die Zahl groß ist, verwenden Sie Amazon Athena, da es über mehrere S3-Objekte läuft, während S3 Select für jeweils ein Objekt funktioniert. Weitere Informationen zur gemeinsamen Verwendung von Amazon S3 und Athena finden Sie unter [Abfragen von Amazon S3 Inventory mit Amazon Athena](#) und [Using Athena \(Verwendung von Athena\)](#) Im Blogbeitrag [Encrypting objects with Amazon S3 Batch Operations](#).

So filtern Sie Ihren S3-Bestandsbericht mithilfe von S3 Select

1. Öffnen Sie die `manifest.json`-Datei aus Ihrem Bestandsbericht und schauen Sie sich den `fileSchema`-Abschnitt des JSON an. Dadurch wird die Abfrage informiert, die Sie für die Daten ausführen.

Der folgende JSON ist eine Beispiels-`manifest.json`-Datei für einen CSV-formatierten Bestand in einem Bucket mit aktivierter Versionierung. Je nachdem, wie Sie Ihren Bestandsbericht konfiguriert haben, sieht Ihr Manifest möglicherweise anders aus.

```
{
  "sourceBucket": "batchoperationsdemo",
  "destinationBucket": "arn:aws:s3:::testbucket",
  "version": "2021-05-22",
  "creationTimestamp": "1558656000000",
  "fileFormat": "CSV",
  "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
BucketKeyStatus",
  "files": [
    {
      "key": "demoinv/batchoperationsdemo/DemoInventory/data/009a40e4-
f053-4c16-8c75-6100f8892202.csv.gz",
      "size": 72691,
      "MD5checksum": "c24c831717a099f0ebe4a9d1c5d3935c"
    }
  ]
}
```

Wenn die Versionierung für den Bucket nicht aktiviert ist oder wenn Sie den Bericht für die neuesten Versionen ausführen möchten, ist die `fileSchema` `Bucket, Key, und BucketKeyStatus`.

Wenn die Versionsverwaltung aktiviert ist, beinhaltet das fileSchema je nach der Einrichtung des Bestandsberichts möglicherweise Folgendes: Bucket, Key, VersionId, IsLatest, IsDeleteMarker, BucketKeyStatus. Achten Sie also auf die Spalten 1, 2, 3 und 6, wenn Sie Ihre Abfrage ausführen.

S3 Batch Operations benötigt den Bucket, den Schlüssel und die Versions-ID als Eingabe, um den Auftrag auszuführen, zusätzlich zu dem Feld, nach dem gesucht werden soll, nämlich BucketKeyStatus. Das Versions-ID-Feld wird nicht benötigt. Diese Angabe ist jedoch hilfreich, wenn Sie mit einem versionierten Bucket arbeiten. Weitere Informationen finden Sie unter [Arbeiten mit Objekten in einem versioning-fähigen Bucket](#).

2. Suchen Sie die Datendateien für den Bestandsbericht. Das manifest.json-Objekt listet die Datendateien unter files (Dateien) auf.
3. Klicken Sie nach dem Suchen und Auswählen der Datendatei in der S3-Konsole auf Actions (Aktionen) und wählen Sie dann Query with S3 Select (Abfragen mit S3 Select).
4. Bewahren Sie die Voreinstellung CSV, Komma, und GZIP ausgewählt, und wählen Sie Next (Weiter).
5. Um das Format Ihres Bestands zu überprüfen, bevor Sie fortfahren, wählen Sie Show file preview (Dateivorschau anzeigen).
6. Geben Sie die Spalten, auf die Sie verweisen möchten, in das SQL-Ausdruck-Feld ein und wählen Sie Run SQL (SQL ausführen) aus. Der folgende Ausdruck gibt die Spalten 1 bis 3 für alle Objekte zurück, die keinen S3-Bucket-Schlüssel konfiguriert haben.

```
select s._1, s._2, s._3 from s3object s where s._6 = 'DISABLED'
```

Nachfolgend sehen Sie einige Beispielergebnisse.

```
batchoperationsdemo,0100059%7Ethumb.jpg,lsrtIxksLu0R0ZkYPL.LhgD5caTYn6vu
batchoperationsdemo,0100074%7Ethumb.jpg,sd2M60g6Fdazoi6D5kNARIE7KzUibmHR
batchoperationsdemo,0100075%7Ethumb.jpg,TLYESLn11mXD5c4Bwi0IinqFrktdkkoL
batchoperationsdemo,0200147%7Ethumb.jpg,amufzfMi_fEw0Rs99rxR_HrDF1E.l3Y0
batchoperationsdemo,0301420%7Ethumb.jpg,9qGU2SEscL.C.c_sK89trmXYIwooABSh
batchoperationsdemo,0401524%7Ethumb.jpg,ORnEWNuB1QhHrrYAGFsZhbyvEYJ3DUor
batchoperationsdemo,200907200065HQ
%7Ethumb.jpg,d8LgvIVjbDR5mUVwW6pu9ahTfReyn5V4
batchoperationsdemo,200907200076HQ
%7Ethumb.jpg,XUT25d7.gK40u_GmnupdaZg3BVx2jN40
```

```
batchoperationsdemo,201103190002HQ
%7Ethumb.jpg,z.2sVRh0myqVi0BuIrngWlsRPQdb7q0S
```

7. Laden Sie die Ergebnisse herunter, speichern Sie sie in einem CSV-Format und laden Sie sie als Liste der Objekte für den S3-Batchoperations-Auftrag in Amazon S3 hoch.
8. Wenn Sie mehrere Manifestdateien haben, führen Sie auch auf diesen Query with S3 Select (Abfragen mit S3 Select) aus. Abhängig von der Größe der Ergebnisse können Sie die Listen kombinieren und einen einzelnen Auftrag für S3-Batch-Vorgänge ausführen oder jede Liste als separater Auftrag ausführen.

Betrachten Sie den [Preis](#) für jede Ausführung eines S3-Batchoperation-Auftrags, wenn Sie die Anzahl der auszuführenden Aufträge festlegen.

Schritt 3: Einrichten und Ausführen des Auftrags für S3-BatchVorgänge

Nachdem Sie nun Ihre gefilterten CSV-Listen von S3-Objekten haben, können Sie mit dem S3-Batchoperations-Auftrag beginnen, um die Objekte mit S3-Bucket-Schlüssel zu verschlüsseln.

Ein Auftrag bezieht sich kollektiv auf die Liste (Manifeste) der bereitgestellten Objekte, die durchgeführte Operation und die angegebenen Parameter. Die einfachste Möglichkeit zum Verschlüsseln dieses Objektsatzes besteht in der Verwendung der PUT-Copy-Operation mit der gleichen Zielpräfix wie die im Manifest aufgeführten Objekte. Dadurch werden entweder die vorhandenen Objekte in einem nicht versionierten Bucket überschrieben, oder bei aktivierter Versionierung wird eine neuere, verschlüsselte Version der Objekte erstellt.

Geben Sie beim Kopieren der Objekte an, dass Amazon S3 das Objekt mit SSE-KMS-Verschlüsselung und S3 verschlüsseln soll. Dieser Auftrag kopiert die Objekte, sodass alle Objekte nach Fertigstellung ein aktualisiertes Erstellungsdatum anzeigen, unabhängig davon, wann Sie sie ursprünglich zu S3 hinzugefügt haben. Geben Sie auch die anderen Eigenschaften für Ihre Gruppe von Objekten als Teil des S3 Batchoperations-Auftrags an, einschließlich Objekt-Markierungen und Speicherklasse.

Teilschritte

- [Einrichten Ihrer IAM-Richtlinie](#)
- [IAM-Rolle für Batch-Vorgänge einrichten](#)
- [Aktivieren von S3-Bucket-Schlüssel für einen vorhandenen Bucket](#)
- [Erstellen eines Auftrags für S3-BatchVorgänge](#)
- [Ausführen eines Auftrags für S3-BatchVorgänge](#)

- [Wissenswertes](#)

Einrichten Ihrer IAM-Richtlinie

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Policy (Richtlinie) und dann Create Policy (Richtlinie erstellen).
3. Wählen Sie den Tab JSON. Klicken Sie auf Edit policy (Richtlinie bearbeiten) und fügen Sie die Beispielrichtlinie für IAM hinzu, die im folgenden Codeblock angezeigt wird.

Nachdem Sie das Richtlinienbeispiel in Ihre [IAM-Konsole](#) kopiert haben, ersetzen Sie Folgendes:

- a. Ersetzen Sie *SOURCE_BUCKET_FOR_COPY* durch den Namen Ihres Quell-Buckets.
- b. Ersetzen Sie *DESTINATION_BUCKET_FOR_COPY* durch den Namen Ihres Ziel-Buckets.
- c. Ersetzen Sie *MANIFEST_KEY* durch den Namen Ihres Manifest-Objekts.
- d. Ersetzen Sie *REPORT_BUCKET* durch den Namen des Buckets, in dem die Berichte gespeichert werden sollen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CopyObjectsToEncrypt",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectTagging",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectVersionAcl",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::SOURCE_BUCKET_FOR_COPY/*",
```

```
    "arn:aws:s3:::DESTINATION_BUCKET_FOR_COPY/*"
  ],
},
{
  "Sid": "ReadManifest",
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion"
  ],
  "Resource": "arn:aws:s3:::MANIFEST_KEY"
},
{
  "Sid": "WriteReport",
  "Effect": "Allow",
  "Action": [
    "s3:PutObject"
  ],
  "Resource": "arn:aws:s3:::REPORT_BUCKET/*"
}
]
}
```

4. Wählen Sie Next: Markierungen (Weiter: Markierungen) aus.
5. Wählen Sie alle gewünschten Markierungen aus (optional) und wählen Sie Next: Review (Weiter: Überprüfung).
6. Geben Sie einen Namen und optional eine Beschreibung für die Richtlinie an und wählen Sie dann Create policy (Richtlinie erstellen) aus.
7. Wählen Sie Review policy (Richtlinie überprüfen) aus und klicken Sie anschließend auf Save changes (Änderungen speichern).
8. Wenn Ihre S3-Batchoperations-Richtlinie jetzt abgeschlossen ist, bringt die Konsole Sie zur Seite der IAM-Richtlinien zurück. Markieren Sie den Richtliniennamen, klicken Sie auf die Schaltfläche links neben dem Richtliniennamen und wählen Sie Policy actions (Richtlinienaktionen), und dann Attach (Hinzufügen).

Um die neu erstellte Richtlinie einer IAM-Rolle zuzuordnen, wählen Sie die entsprechenden Benutzer, Gruppen oder Rollen in Ihrem Konto aus und wählen Sie Attach policy (Richtlinie anfügen) aus. Das bringt Sie zurück zur IAM-Konsole.

IAM-Rolle für Batch-Vorgänge einrichten

1. Wählen Sie im Navigationsbereich der [IAM-Konsole](#) Rollen und dann Rolle erstellen aus.
2. Wählen Sie AWS-Service, S3 und S3 Batch Operations. Wählen Sie dann Next: Permissions aus.
3. Beginnen Sie mit der Eingabe des Namens der IAM-Richtlinie, die Sie soeben erstellt haben. Aktivieren Sie das Kontrollkästchen nach dem Richtliniennamen, wenn es angezeigt wird, und wählen Sie Next: Markierungen (Weiter: Markierungen).
4. (Optional) Fügen Sie Markierungen hinzu, oder lassen Sie die Schlüssel- und Wertefelder für diese Übung leer. Klicken Sie auf Next: Review (Weiter: Prüfen).
5. Geben Sie einen Rollennamen ein, akzeptieren Sie die Standard-Beschreibung oder fügen Sie eine eigene hinzu. Wählen Sie Create role aus.
6. Stellen Sie sicher, dass der Benutzer, der den Auftrag erstellt, über die Berechtigungen im folgenden Beispiel verfügt.

Ersetzen Sie `{ACCOUNT-ID}` durch Ihre AWS-Konto-ID und `{IAM_ROLE_NAME}` durch den Namen, den Sie auf die IAM-Rolle anwenden möchten, die Sie später während der Auftragserstellung in Batch Operations erstellen werden. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen für Amazon-S3-BatchVorgänge](#).

```
{
  "Sid": "AddIamPermissions",
  "Effect": "Allow",
  "Action": [
    "iam:GetRole",
    "iam:PassRole"
  ],
  "Resource": "arn:aws:iam::ACCOUNT-ID:role/IAM_ROLE_NAME"
}
```

Aktivieren von S3-Bucket-Schlüssel für einen vorhandenen Bucket

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Bucket aus, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.


5. Unter Verschlüsselungstyp können Sie zwischen Von Amazon S3 verwalteten Schlüsseln (SSE-S3) und AWS Key Management Service-Schlüsseln (SSE-KMS) wählen.
6. Wenn Sie AWS Key Management Service-Schlüssel (SSE-KMS) ausgewählt haben, können Sie unter AWS KMS key mithilfe einer der folgenden Optionen den AWS KMS-Schlüssel angeben.
 - Wenn Sie aus einer Liste verfügbarer KMS-Schlüssel auswählen möchten, wählen Sie Aus Ihren AWS KMS-Schlüsseln wählen aus. Wählen Sie in der Liste verfügbarer Schlüssel einen KMS-Schlüssel mit symmetrischer Verschlüsselung aus, der sich in derselben Region wie Ihr Bucket befindet. Sowohl der von AWS verwaltete Schlüssel (aws/s3) als auch Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt.
 - Wählen Sie zum Eingeben des KMS-Schlüssel-ARN AWS KMS-Schlüssel-ARN eingeben aus und geben Sie Ihren KMS-Schlüssel-ARN in das angezeigte Feld ein.
 - Wählen Sie zum Erstellen eines neuen vom Kunden verwalteten Schlüssels in der AWS KMS-Konsole Erstellen eines KMS-Schlüssels aus.
7. Unter Bucket Key (Bucket-Schlüssel), wählen Sie Enable (Aktivieren) und dann Save changes (Änderungen speichern) aus.

Nachdem nun ein S3-Bucket-Schlüssel auf Bucket-Ebene aktiviert ist, übernehmen Objekte, die in diesen Bucket hochgeladen, geändert oder kopiert werden, diese Verschlüsselungskonfiguration standardmäßig. Dies umfasst Objekte, die mit Amazon S3 Batch Operations kopiert wurden.

Erstellen eines Auftrags für S3-BatchVorgänge

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich die Option Batch Operations (Batch-Vorgänge) und dann Create Job (Auftrag erstellen).
3. Wählen Sie die Region, in der Sie Ihre Objekte speichern und wählen Sie CSV als Manifest-Typ.
4. Geben Sie den Pfad ein, oder navigieren Sie zu der CSV-Manifestdatei, die Sie zuvor aus Ergebnisse von S3 Select (oder Athena) erstellt haben. Wenn Ihr Manifest Versions-IDs enthält, aktivieren Sie dieses Kontrollkästchen. Wählen Sie Next.
5. Wählen Sie die Copy-Operation und wählen Sie den Ziel-Bucket für den Kopiervorgang aus. Sie können serverseitige Verschlüsselung deaktiviert lassen. Solange S3 Bucket-Schlüssel für den Ziel-Bucket aktiviert ist, wendet der Kopiervorgang S3 Bucket-Schlüssel auf den Ziel-Bucket an.
6. (Optional) Wählen Sie nach Bedarf eine Speicherklasse und die anderen Parameter aus. Die in diesem Schritt angegebenen Parameter gelten für alle Vorgänge, die an den im Manifest aufgeführten Objekten ausgeführt werden. Wählen Sie Weiter aus.

7. Gehen Sie zum Konfigurieren der serverseitigen Verschlüsselung wie folgt vor:
 - a. Wählen Sie unter Serverseitige Verschlüsselung eine der folgenden Optionen aus:
 - Um die Bucket-Einstellungen für die serverseitige Standardverschlüsselung von Objekten beizubehalten, wenn sie in Amazon S3 gespeichert werden, wählen Sie Keinen Verschlüsselungsschlüssel angeben aus. Solange S3-Bucket-Schlüssel für den Ziel-Bucket aktiviert sind, wendet der Kopiervorgang S3-Bucket-Schlüssel auf den Ziel-Bucket an.
 - b. Wenn Sie unter Verschlüsselungseinstellungen die Option Verschlüsselungsschlüssel angeben auswählen, müssen Sie entweder Verwenden von Ziel-Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Ziel-Bucket-Einstellungen für die Standardverschlüsselung auswählen.
 - c. Wenn Sie Überschreiben der Ziel-Bucket-Einstellungen für die Standardverschlüsselung auswählen, müssen Sie die folgenden Verschlüsselungseinstellungen konfigurieren.
 - i. Unter Verschlüsselungstyp müssen Sie entweder Von Amazon S3 verwaltete Schlüssel (SSE-S3) oder AWS Key Management Service-Schlüssel (SSE-KMS) auswählen. SSE-S3 verwendet für die Verschlüsselung der einzelnen Objekte eine der stärksten Blockverschlüsselungen: 256-bit Advanced Encryption Standard (AES-256). Mit SSE-KMS erhalten Sie mehr Kontrolle über Ihren Schlüssel. Weitere Informationen finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#) und [Verwenden Sie serverseitige Verschlüsselung mit AWS KMS Schlüssel \(SSE-KMS\)](#).
 - ii. Wenn Sie AWS Key Management Service-Schlüssel (SSE-KMS) auswählen, können Sie unter AWS KMS key mithilfe einer der folgenden Optionen Ihren AWS KMS key angeben.

 Note

Wenn die Bucket-Richtlinie für das angegebene Ziel vorschreibt, dass Objekte verschlüsselt werden müssen, bevor sie in Amazon S3 gespeichert werden, müssen Sie einen Verschlüsselungsschlüssel angeben. Andernfalls schlägt das Kopieren von Objekten in das Ziel fehl.

- Wenn Sie aus einer Liste verfügbarer KMS-Schlüssel auswählen möchten, wählen Sie Aus Ihren AWS KMS keys wählen und anschließend einen KMS-Schlüssel mit symmetrischer Verschlüsselung aus, der sich in derselben Region wie Ihr Bucket befindet. Sowohl der von AWS verwaltete Schlüssel (aws/s3) als auch Ihre vom Kunden verwalteten Schlüssel werden in dieser Liste angezeigt.
 - Wählen Sie zum Eingeben des KMS-Schlüssel-ARN AWS KMS-Schlüssel-ARN eingeben aus und geben Sie Ihren KMS-Schlüssel-ARN in das angezeigte Feld ein.
 - Wählen Sie zum Erstellen eines neuen vom Kunden verwalteten Schlüssels in der AWS KMS-Konsole Erstellen eines KMS-Schlüssels aus.
- iii. Wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren). Die Copy-Operation wendet einen S3-Bucket-Schlüssel auf den Ziel-Bucket an.
8. Geben Sie Ihrem Job eine Beschreibung (oder behalten Sie die Standardeinstellung bei), legen Sie die Prioritätsstufe fest, wählen Sie einen Berichtstyp aus und geben Sie den Pfad zum Ziel des Fertigstellungsberichts an.
9. Stellen Sie im Abschnitt Permissions (Berechtigungen) sicher, dass Sie die IAM-Rolle für Batch-Vorgänge auswählen, die Sie zuvor definiert haben. Wählen Sie Next.
10. Überprüfen Sie die Einstellungen unter Review (Überprüfen). Wenn Sie Änderungen vornehmen müssen, wählen Sie Previous (Vorherige) aus. Nachdem Sie die Einstellungen für Batch-Vorgänge bestätigt haben, wählen Sie Create Job (Auftrag erstellen).

Weitere Informationen finden Sie unter [Erstellen eines S3-Batch-Vorgangsauftrags](#).

Ausführen eines Auftrags für S3-BatchVorgänge

Der Setup-Assistent kehrt Sie automatisch zum Abschnitt S3-Batch-Vorgänge der Amazon S3-Konsole zurück. Ihr neuer Auftrag wechselt vom New-Zustand in den Preparing-Zustand, da S3 den Prozess beginnt. Während des Preparing-Zustands liest S3 das Manifest des Auftrags, prüft es auf Fehler und berechnet die Anzahl der Objekte.

1. Klicken Sie auf die Aktualisierungs-Schaltfläche in der Amazon S3-Konsole, um den Fortschritt zu überprüfen. Abhängig von der Größe des Manifests kann das Lesen Minuten oder Stunden dauern.
2. Nachdem S3 das Manifest des Auftrags gelesen hat, wechselt der Auftrag in den Zustand Awaiting your confirmation (Wartet auf Ihre Bestätigung). Klicken Sie auf die Options-Schaltfläche links neben der Auftrags-ID und wählen Sie Run job (Auftrag ausführen).

3. Wählen Sie die Einstellungen für den Auftrag aus und wählen Sie in der unteren rechten Ecke des Fensters Run job (Auftrag ausführen) aus.

Nachdem der Auftrag ausgeführt wird, können Sie die Schaltfläche Aktualisieren auswählen, um den Fortschritt über die Dashboard-Ansicht der Konsole zu überprüfen oder den bestimmten Auftrag auszuwählen.

4. Wenn der Auftrag abgeschlossen wurde, können Sie die Successful (erfolgreichen) und Failed (fehlgeschlagenen) Objektzählungen anzeigen, um zu bestätigen, dass alles wie erwartet ausgeführt wurde. Wenn Sie Auftragsberichte aktiviert haben, überprüfen Sie Ihren Auftragsbericht auf die genaue Ursache für fehlgeschlagene Vorgänge.

Sie können diese Schritte auch mithilfe der AWS CLI, AWS SDKs oder Amazon-S3-REST-API ausführen. Weitere Informationen zum Nachverfolgen von Auftragsstatus- und Abschlussberichten finden Sie unter [Verfolgen von Auftragsstatus- und Abschluss](#).

Wissenswertes

Berücksichtigen Sie die folgenden Probleme, wenn Sie S3 Batch Operations verwenden, um Objekte mit S3-Bucket-Schlüsseln zu verschlüsseln:

- Die Kosten für Aufträge, Objekte und Anforderungen in S3 Batch Operations werden Ihnen zusätzlich zu allen mit der Operation, die S3 Batch Operations in Ihrem Namen ausführt, verbundenen Kosten berechnet, einschließlich Datenübertragungen, Anforderungen und anderen Gebühren. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).
- Wenn Sie einen versionierten Bucket verwenden, erstellt jeder ausgeführte S3-Batchoperations-Auftrag neue verschlüsselte Versionen Ihrer Objekte. Darüber hinaus werden die vorherigen Versionen ohne konfigurierten S3-Bucket-Schlüssel beibehalten. Richten Sie zum Löschen der alten Versionen eine S3-Lebenszyklus-Ablauf-Richtlinie für nicht aktuelle Versionen ein, wie unter [Elemente der Lebenszyklus-Konfiguration](#) beschrieben.
- Mit dem Kopiervorgang werden neue Objekte mit neuen Erstellungsdaten erstellt, die sich auf Lebenszyklus-Aktionen wie die Archivierung auswirken können. Wenn Sie alle Objekte in Ihrem Bucket kopieren, weisen alle neuen Kopien identische oder ähnliche Erstellungsdaten auf. Um diese Objekte weiter zu identifizieren und unterschiedliche Lebenszyklus-Regeln für verschiedene Daten-Teilmengen zu erstellen, sollten Sie Objekt-Markierungen verwenden.

Übersicht

In diesem Abschnitt sortieren Sie vorhandene Objekte, um bereits verschlüsselte Daten herauszufiltern. Anschließend haben Sie die Funktion S3-Bucket-Schlüssel auf unverschlüsselte Objekte angewendet, indem Sie mithilfe von S3 Batch Operations vorhandene Daten in einen Bucket mit aktiviertem S3-Bucket-Schlüssel kopiert haben. Dieser Prozess kann Ihnen Zeit und Geld sparen, während Sie Vorgänge wie die Verschlüsselung aller vorhandenen Objekte abschließen können.

Weitere Informationen über S3-Batch-Vorgänge finden Sie unter [Ausführung umfangreicher Batch-Vorgänge für Amazon S3-Objekte durch.](#)

Beispiele, die den Kopiervorgang mit Markierungen unter Verwendung der AWS CLI und AWS SDK for Java zeigen finden Sie unter [Erstellen eines Batchoperations-Auftrags mit Auftrags-Markierungen zur Kennzeichnung.](#)

Aufrufen AWS Lambda function

Die Invoke AWS Lambda Funktion initiiert AWS Lambda Funktionen zum Ausführen benutzerdefinierter Aktionen für Objekte, die in einem Manifest aufgeführt sind. In diesem Abschnitt wird beschrieben, wie Sie eine Lambda-Funktion zur Verwendung mit S3-Batch-Vorgänge erstellen und wie Sie einen Auftrag zum Aufrufen der Funktion erstellen. Die S3-Batch-Vorgängeaufgabe verwendet die LambdaInvoke-Operation, um eine Lambda-Funktion für jedes in einem Manifest aufgeführte Objekt auszuführen.

Sie können mit S3 Batch Operations for Lambda arbeiten, indem Sie AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDKs, oder REST APIs. Weitere Informationen zur Verwendung von Lambda finden Sie unter [Erste Schritte mit AWS Lambda](#) in der AWS Lambda Leitfaden für Entwickler.

In den folgenden Abschnitten werden die ersten Schritte mit S3-Batch-Vorgänge mit Lambda erläutert.

Themen

- [Verwenden von Lambda mit Amazon S3-BatchVorgänge](#)
- [Erstellen einer Lambda-Funktion zur Verwendung mit S3-BatchVorgänge](#)
- [Erstellen eines S3-Batchoperations-Auftrags, der eine Lambda-Funktion aufruft](#)
- [Bereitstellen von Informationen auf Aufgabenebene in Lambda-Manifesten](#)
- [Lernen mit einem Tutorial zu S3-Batchoperationen](#)

Verwenden von Lambda mit Amazon S3-Batchvorgänge

Bei Verwendung von S3 Batch Operations mit AWS Lambda, müssen Sie neue Lambda-Funktionen speziell für die Verwendung mit S3 Batch Operations erstellen. Sie können vorhandene ereignisbasierte Funktionen von Amazon S3 nicht mit S3-Batch-Vorgänge wiederverwenden. Ereignisfunktionen können Nachrichten lediglich erhalten und nicht zurückgeben. Die Lambda-Funktionen, die mit S3-Batch-Vorgänge verwendet werden, müssen Nachrichten akzeptieren und zurückgeben. Weitere Informationen zur Verwendung von Lambda mit Amazon S3 S3-Ereignissen finden Sie unter [Verwenden AWS Lambda mit Amazon S3](#) in der AWS Lambda Leitfaden für Entwickler.

Sie erstellen einen S3-Batchoperations-Auftrag, der Ihre Lambda-Funktion aufruft. Der Auftrag führt dieselbe Lambda-Funktion für alle in Ihrem Manifest aufgeführten Objekte aus. Sie können steuern, welche Versionen Ihrer Lambda-Funktion während der Verarbeitung der Objekte in Ihrem Manifest verwendet werden sollen. S3 Batch Operations unterstützt unqualifizierte Amazon-Ressourcennamen (ARNs), Aliase und bestimmte Versionen. Weitere Informationen finden Sie unter Einführung in [AWS Lambda Versionierung](#) in der AWS Lambda Leitfaden für Entwickler.

Wenn Sie dem S3 Batch Operations-Job eine Funktion zur Verfügung stellen ARN, die einen Alias oder den \$LATEST Qualifier verwendet, und Sie die Version aktualisieren, auf die einer dieser beiden verweist, beginnt S3 Batch Operations, die neue Version Ihrer Lambda-Funktion aufzurufen. Dies kann nützlich sein, wenn Sie mitten in der Ausführung einer großen Aufgabe Funktionen aktualisieren möchten. Wenn S3-Batch-Vorgänge die verwendete Version nicht ändern soll, geben Sie die spezifische Version im Parameter `FunctionARN` an, wenn Sie den Auftrag erstellen.

Verwenden von Lambda und Amazon-S3-Stapelvorgängen mit Verzeichnis-Buckets.

Verzeichnis-Buckets sind eine Art von Amazon-S3-Buckets, die für Workloads oder leistungskritische Anwendungen entwickelt wurden, die eine konstante Latenzzeit im einstelligen Millisekundenbereich erfordern. Weitere Informationen finden Sie unter [Verzeichnis-Buckets](#).

Für die Verwendung von Amazon-S3-Stapelvorgängen zum Aufrufen von Lambda-Funktionen, die auf Verzeichnis-Buckets wirken, gelten besondere Anforderungen. Beispielsweise müssen Sie Ihre Lambda-Anfrage mithilfe eines aktualisierten JSON Schemas strukturieren und angeben, [InvocationSchemaVersion 2.0](#) wann Sie den Job erstellen. Mit diesem aktualisierten Schema können Sie optionale Schlüssel-Wert-Paare für [UserArguments](#) angeben, mit denen Sie bestimmte Parameter vorhandener Lambda-Funktionen ändern können. Weitere Informationen finden Sie unter [Automatisieren der Objektverarbeitung in Amazon S3 S3-Verzeichnis-Buckets mit S3 Batch Operations und AWS Lambda](#) in der AWS Speicher-Blog.

Antwort- und Ergebniscode

S3 Batch Operations ruft die Lambda-Funktion mit einem oder mehreren Schlüsseln auf, denen jeweils ein Schlüssel TaskID zugeordnet ist. S3 Batch Operations erwartet von Lambda-Funktionen einen Ergebniscode pro Schlüssel. Jede in der Anfrage IDs gesendete Aufgabe, die nicht mit einem Ergebniscode pro Schlüssel zurückgegeben wird, erhält den Ergebniscode aus dem Feld. `treatMissingKeysAs` `treatMissingKeysAs` ist ein optionales Anforderungsfeld und hat standardmäßig den Wert. `TemporaryFailure` Die folgende Tabelle enthält die anderen möglichen Ergebniscode und Werte für das `treatMissingKeysAs` Feld.

Antwortcode	Beschreibung
<code>Succeeded</code>	Die Aufgabe wurde normal abgeschlossen. Wenn Sie einen Aufgabenabschlussbericht angefordert haben, ist die Ergebniszeichenfolge der Aufgabe in dem Bericht enthalten.
<code>TemporaryFailure</code>	Die Aufgabe unterlag einem vorübergehenden Fehler und wird erneut ausgeführt, bevor der Job abgeschlossen ist. Die Ergebniszeichenfolge wird ignoriert. Wenn dies der letzte Versuch ist, ist die Fehlermeldung im abschließenden Bericht enthalten.
<code>PermanentFailure</code>	Die Aufgabe unterlag einem dauerhaften Fehler. Wenn Sie einen Aufgabenabschlussbericht angefordert haben, ist die Aufgabe als <code>Failed</code> markiert und enthält die Fehlermeldungszeichenfolge. Ergebniszeichenfolge aus fehlgeschlagenen Aufgaben werden ignoriert.

Erstellen einer Lambda-Funktion zur Verwendung mit S3-Batchvorgänge

Dieser Abschnitt enthält ein Beispiel AWS Identity and Access Management (IAM) Berechtigungen, die Sie mit Ihrer Lambda-Funktion verwenden müssen. Er enthält auch eine Beispiel-Lambda-Funktion zur Verwendung mit S3-Batchvorgänge. Wenn Sie noch nie eine Lambda-Funktion erstellt

haben, finden Sie weitere Informationen unter [Tutorial: Verwenden AWS Lambda mit Amazon S3](#) in der AWS Lambda Leitfaden für Entwickler.

Sie müssen Lambda-Funktionen speziell für die Verwendung mit S3-Batch-Vorgänge erstellen. Sie können vorhandene ereignisbasierte Amazon S3-Lambda-Funktionen nicht wiederverwenden. Dies liegt daran, dass Lambda-Funktionen, die für S3-Batch-Vorgänge verwendet werden, spezielle Datenfelder akzeptieren und zurückgeben müssen.

Important

AWS Lambda In Java geschriebene Funktionen akzeptieren entweder [RequestHandler](#) oder [RequestStreamHandler](#) Handler-Schnittstellen. Um jedoch das Anforderungs- und Antwortformat von S3 Batch Operations zu unterstützen, AWS Lambda benötigt die `RequestStreamHandler` Schnittstelle für die benutzerdefinierte Serialisierung und Deserialisierung einer Anfrage und Antwort. Diese Schnittstelle ermöglicht es Lambda, ein `InputStream` und an die `handleRequest` Java-Methode `OutputStream` zu übergeben. Achten Sie darauf, die `RequestStreamHandler`-Schnittstelle zu verwenden, wenn Sie Lambda-Funktionen mit S3-Batch-Vorgänge verwenden. Wenn Sie eine `RequestHandler` Schnittstelle verwenden, schlägt der Batch-Job fehl und im Abschlussbericht wird „Ungültig JSON zurückgegeben in Lambda payload“ angezeigt.

Weitere Informationen finden Sie unter [Handler-Schnittstellen](#) im AWS Lambda Benutzerleitfaden.

Beispiele von IAM-Berechtigungen

Im Folgenden finden Sie Beispiele für die IAM Berechtigungen, die für die Verwendung einer Lambda-Funktion mit S3 Batch Operations erforderlich sind.

Example – Vertrauensrichtlinie für S3-BatchVorgänge

Im Folgenden finden Sie ein Beispiel für die Vertrauensrichtlinie, die Sie für die IAM Rolle Batch Operations verwenden können. Diese IAM Rolle wird bei der Erstellung des Jobs angegeben und erteilt Batch Operations die Berechtigung, die IAM Rolle zu übernehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

        "Principal": {
            "Service": "batchoperations.s3.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
}

```

Example — Lambda-Richtlinie IAM

Im Folgenden finden Sie ein Beispiel IAM für eine Richtlinie, die S3 Batch Operations die Erlaubnis erteilt, die Lambda-Funktion aufzurufen und das Eingabemanifest zu lesen.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BatchOperationsLambdaPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject",
        "lambda:InvokeFunction"
      ],
      "Resource": "*"
    }
  ]
}

```

Beispiel: Anforderung und Antwort

Dieser Abschnitt bietet Beispiele für Anforderungen und Antworten für die Lambda-Funktion.

Example Anforderung

Das Folgende ist ein JSON Beispiel für eine Anfrage für die Lambda-Funktion.

```

{
  "invocationSchemaVersion": "1.0",
  "invocationId": "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
  "job": {
    "id": "f3cc4f60-61f6-4a2b-8a21-d07600c373ce"
  }
}

```



```
    },
    "tasks": [
      {
        "taskId": "dGFza2lkZ29lc2hlcmUK",
        "s3Key": "customerImage1.jpg",
        "s3VersionId": "1",
        "s3BucketArn": "arn:aws:s3:us-east-1:0123456788:awsexamplebucket1"
      }
    ]
  }
}
```

Example Antwort

Das Folgende ist ein JSON Beispiel für eine Antwort für die Lambda-Funktion.

```
{
  "invocationSchemaVersion": "1.0",
  "treatMissingKeysAs" : "PermanentFailure",
  "invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
  "results": [
    {
      "taskId": "dGFza2lkZ29lc2hlcmUK",
      "resultCode": "Succeeded",
      "resultString": "[\"Mary Major\", \"John Stiles\"]"
    }
  ]
}
```

Beispiel für eine Lambda-Funktion für S3-BatchVorgänge

Im folgenden Beispiel entfernt Python Lambda eine Löschmarkierung von einem versionierten Objekt.

Wie das Beispiel zeigt, sind Schlüssel aus S3 Batch Operations URL codiert. Um Amazon S3 mit anderen zu verwenden AWS Dienste, es ist wichtig, dass Sie den Schlüssel URL dekodieren, der von S3 Batch Operations übergeben wird.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")
```

```
s3 = boto3.client("s3")

def lambda_handler(event, context):
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
                  to remove.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation. When the result code is TemporaryFailure, S3 retries the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]

    try:
        obj_key = parse.unquote(task["s3Key"], encoding="utf-8")
        obj_version_id = task["s3VersionId"]
        bucket_name = task["s3BucketArn"].split(":")[-1]

        logger.info(
            "Got task: remove delete marker %s from object %s.", obj_version_id,
obj_key
        )

    try:
        # If this call does not raise an error, the object version is not a delete
        # marker and should not be deleted.
        response = s3.head_object(
            Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
        )
        result_code = "PermanentFailure"
        result_string = (
```

```
        f"Object {obj_key}, ID {obj_version_id} is not " f"a delete marker."
    )

    logger.debug(response)
    logger.warning(result_string)
except ClientError as error:
    delete_marker = error.response["ResponseMetadata"]["HTTPHeaders"].get(
        "x-amz-delete-marker", "false"
    )
    if delete_marker == "true":
        logger.info(
            "Object %s, version %s is a delete marker.", obj_key,
obj_version_id
        )
        try:
            s3.delete_object(
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
            )
            result_code = "Succeeded"
            result_string = (
                f"Successfully removed delete marker "
                f"{obj_version_id} from object {obj_key}."
            )
            logger.info(result_string)
        except ClientError as error:
            # Mark request timeout as a temporary failure so it will be
retried.

            if error.response["Error"]["Code"] == "RequestTimeout":
                result_code = "TemporaryFailure"
                result_string = (
                    f"Attempt to remove delete marker from "
                    f"object {obj_key} timed out."
                )
                logger.info(result_string)
            else:
                raise
    else:
        raise ValueError(
            f"The x-amz-delete-marker header is either not "
            f"present or is not 'true'."
        )
except Exception as error:
    # Mark all other exceptions as permanent failures.
    result_code = "PermanentFailure"
```

```
    result_string = str(error)
    logger.exception(error)
finally:
    results.append(
        {
            "taskId": task_id,
            "resultCode": result_code,
            "resultString": result_string,
        }
    )
return {
    "invocationSchemaVersion": invocation_schema_version,
    "treatMissingKeysAs": "PermanentFailure",
    "invocationId": invocation_id,
    "results": results,
}
```

Erstellen eines S3-Batchoperations-Auftrags, der eine Lambda-Funktion aufruft

Wenn Sie einen S3-Batchoperations-Auftrag erstellen, um eine Lambda-Funktion aufzurufen, müssen Sie Folgendes angeben:

- Die ARN Ihrer Lambda-Funktion (die den Funktionsalias oder eine bestimmte Versionsnummer enthalten kann)
- Eine IAM Rolle mit der Berechtigung, die Funktion aufzurufen
- Den Aktionsparameter `LambdaInvokeFunction`

Weitere Informationen zum Erstellen eines S3-Batchoperations-Auftrags finden Sie unter [Erstellen eines S3-Batch-Vorgangsauftrags](#) und [Von S3 Batch-Vorgänge unterstützte Vorgänge](#).

Im folgenden Beispiel wird ein S3-Auftrag für Batch Operations erstellt, der eine Lambda-Funktion aufruft, indem AWS CLI.

```
aws s3control create-job
  --account-id <AccountID>
  --operation '{"LambdaInvoke": { "FunctionArn":
    "arn:aws:lambda:Region:AccountID:function:LambdaFunctionName" } }'
```

```
--manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":
{"ObjectArn":"arn:aws:s3:::ManifestLocation","ETag":"ManifestETag"}}'
--report
'{"Bucket":"arn:aws:s3:::awsexamplebucket1","Format":"Report_CSV_20180820","Enabled":true,"Pre
--priority 2
--role-arn arn:aws:iam::AccountID:role/BatchOperationsRole
--region Region
--description "Lambda Function"
```

Bereitstellen von Informationen auf Aufgabenebene in Lambda-Manifesten

Wenn Sie verwenden AWS Lambda funktioniert mit S3 Batch Operations. Möglicherweise möchten Sie, dass zusätzliche Daten zu jeder Aufgabe/Taste, die bearbeitet wird, hinzugefügt werden. Möglicherweise möchten Sie, dass sowohl ein Quellobjektschlüssel als auch ein neuer Objektschlüssel bereitgestellt wird. Ihre Lambda-Funktion kann in einem solchen Fall den Quellschlüssel in einen neuen S3-Bucket unter einem neuen Namen speichern. Standardmäßig können Sie mit Amazon S3-Batch-Vorgänge nur den Ziel-Bucket und eine Liste von Quellschlüsseln im Eingabemanifest für Ihren Auftrag angeben. Nachfolgend wird beschrieben, wie Sie weitere Daten in Ihr Manifest aufnehmen können, sodass Sie komplexere Lambda-Funktionen ausführen können.

Verwenden Sie das folgende URL -kodierte Format, um Schlüsselparameter in Ihrem S3-Batch-Operations-Manifest anzugeben, die im Code Ihrer Lambda-Funktion verwendet werden sollen. JSON Das key-Feld wird an Ihre Lambda-Funktion übergeben, als wäre es ein Amazon S3-Objektschlüssel. Es kann aber von der Lambda-Funktion so interpretiert werden, dass es andere Werte oder mehrere Schlüssel enthält, wie im Folgenden gezeigt:

Note

Die maximale Anzahl von Zeichen für das Feld key im Manifest beträgt 1.024.

Example — Manifest, das die „Amazon S3 S3-Schlüssel“ durch Zeichenketten ersetzt JSON

Die URL -kodierte Version muss für S3 Batch Operations bereitgestellt werden.

```
my-bucket,{"origKey": "object1key", "newKey": "newObject1Key"}
my-bucket,{"origKey": "object2key", "newKey": "newObject2Key"}
my-bucket,{"origKey": "object3key", "newKey": "newObject3Key"}
```

Example — Manifest -kodierte URL

Diese URL -kodierte Version muss für S3 Batch Operations bereitgestellt werden. Die nicht URL -kodierte Version funktioniert nicht.

```
my-bucket,%7B%22origKey%22%3A%20%22object1key%22%2C%20%22newKey%22%3A%20%22newObject1Key%22%7D%20%22object2key%22%2C%20%22newKey%22%3A%20%22newObject2Key%22%7D%20%22object3key%22%2C%20%22newKey%22%3A%20%22newObject3Key%22%7D
```

Example – Lambda-Funktion im Manifest-Format, die Ergebnisse in den Auftragsbericht schreibt.

Dieses URL -kodierte Manifestbeispiel enthält durch Pipen getrennte Objektschlüssel für die folgende Lambda-Funktion zum Analysieren.

```
my-bucket,object1key%7Clower
my-bucket,object2key%7Cupper
my-bucket,object3key%7Creverse
my-bucket,object4key%7Cdelete
```

Diese Lambda-Funktion zeigt, wie eine durch Pipe getrennte Aufgabe analysiert wird, die im Amazon S3-BatchVorgängemanifest codiert ist. Die Aufgabe gibt an, welcher Revisionsvorgang auf das angegebene Objekt angewendet wird.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.resource("s3")

def lambda_handler(event, context):
    """
    Applies the specified revision to the specified object.

    :param event: The Amazon S3 batch event that contains the ID of the object to
```

```
        revise and the revision type to apply.
:param context: Context about the event.
:return: A result structure that Amazon S3 uses to interpret the result of the
        operation.
"""
# Parse job parameters from Amazon S3 batch operations
invocation_id = event["invocationId"]
invocation_schema_version = event["invocationSchemaVersion"]

results = []
result_code = None
result_string = None

task = event["tasks"][0]
task_id = task["taskId"]
# The revision type is packed with the object key as a pipe-delimited string.
obj_key, revision = parse.unquote(task["s3Key"], encoding="utf-8").split("|")
bucket_name = task["s3BucketArn"].split(":")[-1]

logger.info("Got task: apply revision %s to %s.", revision, obj_key)

try:
    stanza_obj = s3.Bucket(bucket_name).Object(obj_key)
    stanza = stanza_obj.get()["Body"].read().decode("utf-8")
    if revision == "lower":
        stanza = stanza.lower()
    elif revision == "upper":
        stanza = stanza.upper()
    elif revision == "reverse":
        stanza = stanza[::-1]
    elif revision == "delete":
        pass
    else:
        raise TypeError(f"Can't handle revision type '{revision}'.")

    if revision == "delete":
        stanza_obj.delete()
        result_string = f"Deleted stanza {stanza_obj.key}."
    else:
        stanza_obj.put(Body=bytes(stanza, "utf-8"))
        result_string = (
            f"Applied revision type '{revision}' to " f"stanza {stanza_obj.key}."
        )
```

```
    logger.info(result_string)
    result_code = "Succeeded"
except ClientError as error:
    if error.response["Error"]["Code"] == "NoSuchKey":
        result_code = "Succeeded"
        result_string = (
            f"Stanza {obj_key} not found, assuming it was deleted "
            f"in an earlier revision."
        )
        logger.info(result_string)
    else:
        result_code = "PermanentFailure"
        result_string = (
            f"Got exception when applying revision type '{revision}' "
            f"to {obj_key}: {error}."
        )
        logger.exception(result_string)
finally:
    results.append(
        {
            "taskId": task_id,
            "resultCode": result_code,
            "resultString": result_string,
        }
    )
return {
    "invocationSchemaVersion": invocation_schema_version,
    "treatMissingKeysAs": "PermanentFailure",
    "invocationId": invocation_id,
    "results": results,
}
```

Lernen mit einem Tutorial zu S3-Batchoperationen

Das folgende Tutorial enthält vollständige end-to-end Verfahren für einige Batch-Operationsaufgaben mit Lambda.

- [Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations, AWS Lambda, und AWS Elemental MediaConvert](#)

Alle Objektmarkierungen ersetzen

Der Vorgang Replace all object tags (Alle Objektmarkierungen ersetzen) ersetzt die Amazon S3-Objektmarkierungen für jedes im Manifest aufgeführte Objekt. Ein Amazon S3-Objekt-Tag ist ein Schlüssel-Wert-Paar, mit dem Sie Metadaten über ein Objekt speichern können.

Um eine Aufgabe „Alle Objektmarkierungen ersetzen“ zu erstellen, geben Sie eine Reihe von Markierungen an, die Sie anwenden möchten. S3 Batch-Vorgänge wendet auf jedes Objekt den gleichen Markierungssatz an. Der von Ihnen bereitgestellte Tag-Satz ersetzt alle Tag-Sätze, die bereits mit den Objekten im Manifest verknüpft sind. S3-Batch-Vorgänge unterstützen nicht das Hinzufügen von Markierungen zu Objekten, während die vorhandenen Markierungen unverändert gelassen werden.

Wenn die Objekte in Ihrem Manifest zu einem versionierten Bucket gehören, können Sie die Tag-Gruppen auf bestimmte Versionen der einzelnen Objekte anwenden. Hierzu geben Sie für jedes Objekt im Manifest eine Versions-ID an. Wenn Sie für ein Objekt keine Versions-ID angeben, wendet S3-Batch-Vorgänge den Tag auf die aktuelle Version jedes Objekts an.

Beschränkungen und Einschränkungen

- Die AWS Identity and Access Management (IAM)-Rolle, die Sie zum Ausführen des Batchoperations-Auftrags angeben, muss über Berechtigungen zum Durchführen der zugrunde liegenden Amazon S3-Operation „Alle Objektmarkierungen ersetzen“ verfügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [PutObjectTagging](#) in der API-Referenz zum Amazon Simple Storage Service.
- S3-Batch-Vorgänge verwenden die Amazon S3 [PutObjectTagging](#)-Operation, um Markierungen auf jedes Objekt im Manifest anzuwenden. Alle Einschränkungen und Begrenzungen, die auf den zugrunde liegenden Vorgang zutreffen, gelten auch für S3-Batchoperations-Aufträge.

Weitere Informationen zur Verwendung der Konsole zum Erstellen von Aufträgen finden Sie unter [Erstellen eines S3-Batchoperationsauftrags](#).

Weitere Informationen zur Objektmarkierung finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#) in diesem Handbuch und unter [PutObjectTagging](#), [GetObjectTagging](#) und [DeleteObjectTagging](#) in der API-Referenz zum Amazon Simple Storage Service.

Alle Objektmarkierungen löschen

Der Vorgang Alle Objekt-Tags löschen entfernt alle Amazon S3-Objekt-Tag-Sätze, die derzeit mit den im Manifest aufgeführten Objekten verknüpft sind. S3 Batch-Vorgänge unterstützt nicht das Löschen von Markierungen aus Objekten, während andere Markierungen beibehalten werden.

Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, können Sie die Tag-Sätze aus einer bestimmten Version eines Objekts entfernen. Hierzu geben Sie für jedes Objekt im Manifest eine Versions-ID an. Wenn Sie keine Versions-ID für ein Objekt angeben, entfernt S3 Batch-Vorgänge den Tag-Satz aus der neuesten Version jedes Objekts.

Weitere Informationen über Batchoperationsmanifeste finden Sie unter [Angeben eines Manifests](#).

Warning

Durch die Ausführung dieser Aufgabe werden alle Objekt-Tag-Sätze für jedes im Manifest aufgelistete Objekt entfernt.

Beschränkungen und Einschränkungen

- Die AWS Identity and Access Management (IAM) Rolle, die Sie zum Ausführen der Aufgabe angeben, muss über Berechtigungen zum Durchführen des zugrunde liegenden Amazon S3-Vorgangs zur Löschung von Objektmarkierungen verfügen. Weitere Informationen finden Sie unter [DeleteObjectTagging](#) in der API-Referenz zum Amazon Simple Storage Service.
- S3 Batch-Vorgänge verwendet den Amazon S3-Vorgang [DeleteObjectTagging](#), um die Markierungssätze von jedem Objekt im Manifest zu entfernen. Alle Einschränkungen und Begrenzungen, die auf den zugrunde liegenden Vorgang zutreffen, gelten auch für S3-Batchoperations-Aufträge.

Informationen zum Erstellen von Aufgaben finden Sie unter [Erstellen eines S3-Batch-Vorgangsauftrags](#).

Weitere Informationen zur Objektmarkierung finden Sie unter [Alle Objektmarkierungen ersetzen](#) in diesem Handbuch und unter [PutObjectTagging](#), [GetObjectTagging](#) und [DeleteObjectTagging](#) in der API-Referenz zum Amazon Simple Storage Service.

Ersetzen der Access Control List

Der Vorgang Access Control List (ACL) ersetzen ersetzt die Amazon S3-Access-Control-Lists (ACLs) für jedes Objekt, das im Manifest aufgeführt ist. Mithilfe von ACLs können Sie festlegen, wer auf ein Objekt zugreifen darf, und welche Aktionen die Person ausführen darf.

S3-Batch-Vorgänge unterstützt benutzerdefinierte ACLs, die von ihnen festgelegt werden, sowie vorgefertigte ACLs, die von Amazon S3 mit vordefinierten Gruppen von Zugriffsberechtigungen bereitgestellt werden.

Wenn die Objekte in Ihrem Manifest zu einem versionierten Bucket gehören, können Sie die ACLs auf bestimmten Versionen der einzelnen Objekte anwenden. Hierzu geben Sie für jedes Objekt im Manifest eine Versions-ID an. Wenn Sie für ein Objekt keine Versions-ID angeben, wendet S3-Batch-Vorgänge die ACL auf die aktuelle Version des Objekts an.

Weitere Informationen zu ACLs in Amazon S3 erhalten Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

S3 Block Public Access

Wenn Sie den öffentlichen Zugriff auf alle Objekte in einem Bucket einschränken möchten, sollten Sie die Amazon S3-Block-Einstellungen für den öffentlichen Zugriff anstelle von S3-Batch-Vorgänge verwenden. Block Public Access kann den öffentlichen Zugriff auf einen Bucket oder ein Konto mithilfe einer einzelnen, einfachen Operation einschränken, die sich zudem auch noch schnell umsetzen lässt. Daher ist dies die bessere Wahl, wenn Sie die Absicht haben, den öffentlichen Zugriff auf alle Objekte in einem Bucket oder einem Konto einzuschränken. Verwenden Sie S3-BatchVorgänge, wenn Sie eine benutzerdefinierte ACL auf jedes Objekt im Manifest anwenden möchten. Weitere Informationen über die S3 Block Public Access finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).

S3 Object Ownership

Wenn sich die Objekte im Manifest in einem Bucket befinden, der die erzwungene Einstellung des Bucket-Eigentümers für „Object Ownership“ verwendet, kann die Operation Replace access control list (ACL) (Zugriffskontrollliste (ACL)) ersetzen nur Objekt-ACLs angeben, die dem Bucket-Eigentümer die volle Kontrolle gewähren. Der Vorgang kann anderen keine Objekt-ACL-Berechtigungen für AWS-Konten oder Gruppen erteilen. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Beschränkungen und Einschränkungen

- Die Rolle, die Sie zum Ausführen der Aufgabe „Access Control List ersetzen“ angeben, muss über Berechtigungen zum Durchführen des zugrunde liegenden Amazon S3-Vorgangs `PutObjectAcl` verfügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [PutObjectAcl](#) in der API-Referenz zum Amazon Simple Storage Service.
- S3-Batch-Vorgänge verwenden die Amazon S3 `PutObjectAcl`-Operation, um die angegebene ACL auf jedes Objekt im Manifest anzuwenden. Daher gelten alle Einschränkungen und Begrenzungen, die auf den zugrunde liegenden `PutObjectAcl`-Vorgang zutreffen, auch für die S3-Batchoperations-Aufträge „Access Control List ersetzen“.

Objekte mit Batch Operations wiederherstellen

Der Wiederherstellungsvorgang initiiert Wiederherstellungsanforderungen für die archivierten Amazon-S3-Objekte, die in Ihrem Manifest aufgeführt sind. Die folgenden archivierten Objekte müssen wiederhergestellt werden, bevor auf sie in Echtzeit zugegriffen werden kann:

- Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert werden
- Objekte, die über die Speicherklasse S3 Intelligent-Tiering in den Stufen Archive Access oder Deep Archive Access archiviert sind

Die Verwendung eines S3 Restore Object-Vorgangs in Ihrem S3 Batchoperations-Auftrag führt zu einer Wiederherstellungs-Anforderung für jedes im Manifest angegebene Objekt.

Important

Die S3-Aufgabe „Initiate Restore Object“ (Objektwiederherstellung initiieren) initiiert nur die Anfrage zum Wiederherstellen von Objekten. S3-Batch-Vorgänge melden die Aufgabe als abgeschlossen für jedes Objekt, nachdem die Anfrage für dieses Objekt initiiert wurde. Amazon S3 aktualisiert den Auftrag nicht und benachrichtigt Sie nicht weiter, wenn die Objekte wiederhergestellt wurden. Sie können jedoch S3-Ereignis-Benachrichtigungen verwenden, um Benachrichtigungen zu erhalten, wenn die Objekte in Amazon S3 verfügbar sind. Weitere Informationen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Wenn Sie die Aufgabe „S3 Initiate Restore Object“ erstellen, sind die folgenden Argumente verfügbar:

ExpirationInDays

Dieses Argument gibt an, wie lange das Objekt S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive in Amazon S3 verfügbar bleibt. Für „Initiate-Restore-Object-Aufgaben, die auf Objekte von S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive ausgelegt sind, muss `ExpirationInDays` auf 1 oder höher festgelegt werden.

Important

Legen Sie `ExpirationInDays` beim Erstellen von „S3 Initiate Restore Object“-Operationsaufgaben, die auf S3 Intelligent-Tiering Archive Access- und Deep Archive Access-Stufenobjekte ausgelegt sind, nicht fest. Objekte in S3 Intelligent-Tiering Archive Access-Stufen unterliegen keinem Ablaufdatum bei der Wiederherstellung, daher führt die Angabe von `ExpirationInDays` zu einem Fehler bei der Wiederherstellungsanforderung.

GlacierJobTier

Amazon S3 kann Objekte mithilfe einer von drei verschiedenen Abrufstufen wiederherstellen: EXPEDITED, STANDARD und BULK. Die S3-Funktion Batch Operations unterstützt jedoch nur die STANDARD Abrufstufen. Weitere Informationen zu den Unterschieden zwischen den Wiederherstellungsstufen finden Sie unter [Archiv-Abrufoptionen](#).

Weitere Informationen zu den Preisen für jede Stufe finden Sie im Abschnitt Anfragen und Datenabrufe auf der Seite [Amazon S3-Preise](#).

Unterschiede bei der Wiederherstellung aus S3 Glacier und S3 Intelligent-Tiering

Das Wiederherstellen archivierter Dateien aus den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive unterscheidet sich von der Wiederherstellung von Dateien aus der Speicherklasse S3 Intelligent-Tiering in den Stufen Archive Access oder Deep Archive Access.

- Wenn Sie von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, wird eine temporäre Kopie des Objekts erstellt. Amazon S3 löscht diese Kopie nachdem der Wert, den Sie im `ExpirationInDays`-Argument angegeben haben, abgelaufen ist. Nachdem diese temporäre Kopie gelöscht wurde, müssen Sie eine zusätzliche Wiederherstellungsanfrage einreichen, um auf das Objekt zugreifen zu können.

- Geben Sie bei der Wiederherstellung archivierter S3 Intelligent-Tiering-Objekte das `ExpirationInDays`-Argument nicht an. Wenn Sie ein Objekt aus den Stufen S3 Intelligent-Tiering Archive Access oder Deep Archive Access wiederherstellen, wird das Objekt zurück in die Speicherklasse S3 Intelligent-Tiering Frequent Access übergehen. Nach mindestens 90 aufeinanderfolgenden Tagen ohne Zugriff wechselt das Objekt automatisch in die Stufe „Archive Access“. Das Objekt wechselt nach mindestens 180 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die Stufe „Deep Archive Access“.
- Batch-Operations-Aufträge können entweder mit Objekte der Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive oder mit Objekten der Speicherstufen S3 Intelligent-Tiering Archive Access und Deep Archive Access arbeiten. Batch Operations kann im selben Auftrag nicht für beide Typen von archivierten Objekten ausgeführt werden. Um Objekte beider Typen wiederherzustellen, müssen Sie separate Batchoperations-Aufgaben erstellen.

Überlappende Wiederherstellungen

Sollte die Aufgabe [S3 Initiate Restore Object](#) (Objektwiederherstellung initiieren) versuchen, ein Objekt wiederherzustellen, das gerade wiederhergestellt wird, geht S3 Batch-Vorgänge folgendermaßen vor:

Der Wiederherstellungsvorgang für das Objekt ist erfolgreich, wenn eine der folgenden Bedingungen erfüllt ist:

- Im Vergleich zu der bereits ausgeführten Wiederherstellungsanforderung ist der `ExpirationInDays`-Wert für diesen Auftrag identisch und sein `GlacierJobTier`-Wert ist schneller.
- Die vorherige Wiederherstellungsanforderung wurde bereits abgeschlossen und das Objekt ist derzeit verfügbar. In diesem Fall aktualisiert Batch-Vorgänge das Ablaufdatum des wiederhergestellten Objekts so, dass es mit dem in der laufenden Wiederherstellungsanforderung angegebenen `ExpirationInDays`-Wert übereinstimmt.

Der Wiederherstellungsvorgang für das Objekt schlägt fehl, wenn eine der folgenden Bedingungen erfüllt sind:

- Die bereits ausgeführte Wiederherstellungsanforderung wurde noch nicht abgeschlossen und die Wiederherstellungsdauer für diese Aufgabe (durch den `ExpirationInDays`-Wert angegeben) unterscheidet sich von der Wiederherstellungsdauer, die in der bereits ausgeführten Wiederherstellungsanforderung angegeben wurde.

- Die Wiederherstellungsebene für diesen Auftrag (durch den `GlacierJobTier`-Wert angegeben) ist identisch oder langsamer als die Wiederherstellungsebene, die in der bereits ausgeführten Wiederherstellungsanforderung angegeben wurde.

Einschränkungen

„S3 Initiate Restore Object“-Aufgaben haben folgende Einschränkungen:

- Sie müssen die Aufgabe in derselben Region wie die archivierten Objekte erstellen.
- S3 Batch-Vorgänge unterstützt die Abrufstufe EXPEDITED nicht.

Weitere Informationen zum Wiederherstellen von Objekten finden Sie unter [Wiederherstellen eines archivierten Objekts](#).

Aufrechterhaltung der S3-Objektsperre

Der Vorgang Objektsperre zur Aufbewahrung ermöglicht es Ihnen, Aufbewahrungsdaten für Ihre Objekte entweder im Governance-Modus oder im Compliance-Modus anzuwenden. Diese Aufbewahrungsmodi wenden unterschiedliche Schutzgrade an. Sie können einen der beiden Aufbewahrungsmodi auf jede Objektversion anwenden. Aufbewahrungsdaten verhindern wie gesetzliche Aufbewahrungsfristen, dass ein Objekt überschrieben oder gelöscht wird. Amazon S3 speichert das in den Metadaten des Objekts angegebene Retain until (Aufbewahren bis)-Datum und schützt die angegebene Version der Objektversion bis zum Ablauf der Aufbewahrungsfrist.

Sie können S3-Batch-Vorgänge mit Objektsperre verwenden, um die Aufbewahrungsdaten vieler Amazon S3-Objekte gleichzeitig zu verwalten. Sie geben die Liste der Zielobjekte in Ihrem Manifest an und senden sie zur Fertigstellung an BatchVorgänge. Weitere Informationen finden Sie unter S3-Objektsperre [the section called “Aufbewahrungszeiträume”](#).

Ihr S3-Batchoperations-Auftrag mit Aufbewahrungsdaten läuft bis zum Abschluss, bis zum Abbruch oder bis ein Fehlerstatus erreicht ist. Sie sollten S3-Batch-Vorgänge und die Aufbewahrung der S3-Objektsperre verwenden, wenn Sie das Aufbewahrungdatum für viele Objekte mit einer einzigen Anfrage hinzufügen, ändern oder entfernen möchten.

Batch-Vorgänge überprüft, ob die Objektsperre in Ihrem Bucket aktiviert ist, bevor Schlüssel im Manifest verarbeitet werden. Um die Vorgänge und die Validierung durchzuführen, benötigt Batch-Vorgänge `s3:GetBucketObjectLockConfiguration`- und `s3:PutObjectRetention`-Berechtigungen in einer IAM-Rolle, damit Batch-Vorgänge die Objektsperre für Sie aufrufen kann. Weitere Informationen finden Sie unter [the section called “Überlegungen zu Object Lock”](#).

Informationen zur Verwendung dieser Operation mit der REST-API finden Sie unter `S3PutObjectRetention` in der [CreateJob](#)-Operation in der Amazon Simple Storage Service API-Referenz.

Ein AWS Command Line Interface-Beispiel für die Verwendung dieser Operation finden Sie unter [the section called "Batch-Vorgänge mit Objektsperrenaufbewahrung verwenden"](#). Ein AWS SDK for Java-Beispiel finden Sie unter [the section called "Batch-Vorgänge mit Objektsperrenaufbewahrung verwenden"](#).

Beschränkungen und Einschränkungen

- S3-Batch-Vorgänge nimmt keine Änderungen auf Bucket-Ebene vor.
- Versionierung und die S3-Objektsperre müssen auf dem Bucket konfiguriert werden, in dem der Auftrag ausgeführt wird.
- Alle im Manifest aufgelisteten Objekte müssen sich im selben Bucket befinden.
- Die Operation funktioniert mit der neuesten Version des Objekts, es sei denn, eine Version ist explizit im Manifest angegeben.
- Sie benötigen eine `s3:PutObjectRetention`-Berechtigung in Ihrer IAM-Rolle, um dies zu verwenden.
- `s3:GetBucketObjectLockConfiguration` Die -IAM-Berechtigung ist erforderlich, um zu bestätigen, dass die Objektsperre für den S3-Bucket aktiviert ist.
- Sie können den Aufbewahrungszeitraum von Objekten nur verlängern, bei denen die Aufbewahrungsdaten des COMPLIANCE-Modus angewendet wurden, und er kann nicht verkürzt werden.

S3-Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen

Die Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen ermöglicht es Ihnen, eine Objektversion legal zu halten. Wie das Festlegen von Aufbewahrungszeiträumen verhindern auch rechtliche Aufbewahrungsfristen das Überschreiben oder Löschen von Objektversionen. Mit rechtlichen Aufbewahrungsfristen sind jedoch keine Aufbewahrungszeiträume verknüpft. Sie sind gültig, bis sie entfernt werden.

Sie können S3-Batch-Vorgänge mit Objektsperre verwenden, um vielen Amazon S3-Objekten gleichzeitig gesetzliche Aufbewahrungsfristen hinzuzufügen. Sie können dies tun, indem Sie die Zielobjekte in Ihrem Manifest auflisten und diese Liste an Batch-Vorgänge senden. Ihr S3-

Batchoperations-Auftrag mit Objektsperre für die gesetzliche Aufbewahrungsfrist bis zum Abschluss, bis zur Kündigung oder bis ein Ausfallstatus erreicht ist, ausgeführt.

S3-Batch-Vorgänge überprüft, ob die Objektsperre in Ihrem S3-Bucket aktiviert ist, bevor Schlüssel im Manifest verarbeitet werden. Um die Objekt-Vorgänge und die Validierung auf Bucket-Ebene durchzuführen, benötigt S3-Batch-Vorgänge `s3:PutObjectLegalHold` und `s3:GetBucketObjectLockConfiguration` in einer IAM-Rolle, die es S3-Batch-Vorgänge ermöglicht, die S3-Objektsperre für Sie aufzurufen.

Wenn Sie den S3-Batchoperations-Auftrag erstellen, um die gesetzliche Aufbewahrungsfrist zu entfernen, müssen Sie lediglich Off (Aus) als Status der gesetzlichen Aufbewahrungsfrist angeben. Weitere Informationen finden Sie unter [the section called "Überlegungen zu Object Lock"](#).

Informationen zur Verwendung dieser Operation mit der REST-API finden Sie unter `S3PutObjectLegalHold` in der [CreateJob](#)-Operation in der API-Referenz zum Amazon Simple Storage Service.

Eine Beispielverwendung dieser Operation finden Sie unter [Verwenden des AWS SDK for Java](#).

Beschränkungen und Einschränkungen

- S3-Batch-Vorgänge nimmt keine Änderungen auf Bucket-Ebene vor.
- Alle im Manifest aufgelisteten Objekte müssen sich im selben Bucket befinden.
- Versionierung und die S3-Objektsperre müssen auf dem Bucket konfiguriert werden, in dem der Auftrag ausgeführt wird.
- Die Operation funktioniert mit der neuesten Version des Objekts, es sei denn, eine Version ist explizit im Manifest angegeben.
- `s3:PutObjectLegalHold` Die -Berechtigung ist in Ihrer IAM-Rolle erforderlich, um Objekten gesetzliche Aufbewahrungsfristen hinzuzufügen oder von diesen zu entfernen.
- `s3:GetBucketObjectLockConfiguration` Die IAM-Berechtigung ist erforderlich, um zu bestätigen, dass die S3-Objektsperre für den S3-Bucket aktiviert ist.
- [Kopieren von Objekten](#)
- [Aufrufen AWS Lambda function](#)
- [Alle Objektmarkierungen ersetzen](#)
- [Alle Objektmarkierungen löschen](#)

- [Ersetzen der Access Control List](#)
- [Objekte mit Batch Operations wiederherstellen](#)
- [Aufrechterhaltung der S3-Objektsperre](#)
- [S3-Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen](#)
- [Replizieren bestehender Objekte mit S3-Batch-Replikation](#)

Verwalten von S3-Batch-Vorgangsaufträgen

Amazon S3 bietet robuste Tools, die Sie bei der Verwaltung Ihrer Aufträge für die S3-Batch-Vorgänge unterstützen, nachdem Sie diese erstellt haben. In diesem Abschnitt werden die Vorgänge beschrieben, mit denen Sie Ihre Aufträge mithilfe der AWS Management Console, AWS CLI, AWS-SDKs oder REST-API verwalten und verfolgen können.

Themen

- [Verwenden der Amazon-S3-Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge](#)
- [Auflisten von Aufträgen](#)
- [Anzeigen von Auftragsdetails](#)
- [Zuweisen der Auftragspriorität](#)

Verwenden der Amazon-S3-Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge

Verwenden der Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge Beispielsweise ist Folgendes möglich:

- Aktive und in Warteschlange gewartete Aufträge anzeigen
- Ändern der Priorität eines Auftrags
- Bestätigen und Ausführen eines Auftrags
- Einen Auftrag klonen
- Abbrechen eines Auftrags

Verwenden der Konsole zum Verwalten Ihrer Batchvorgänge

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie den spezifischen Auftrag aus, den Sie verwalten möchten.

Auflisten von Aufträgen

Sie können eine Liste Ihrer S3-Batch-Vorgangsaufträge abrufen. Die Liste umfasst Aufträge, die noch nicht abgeschlossen sind, sowie Aufträge, die in den letzten 90 Tagen abgeschlossen wurden. Die Auftragsliste enthält Informationen zu jedem Auftrag, z. B. die ID, die Beschreibung, die Priorität, den aktuellen Status und die Anzahl an Aufgaben, die erfolgreich durchgeführt wurden oder fehlgeschlagen sind. Sie können die Auftragsliste nach dem Status filtern. Wenn Sie eine Auftragsliste über die Konsole abrufen, können Sie Ihre Aufträge auch nach Beschreibung oder ID durchsuchen und nach AWS-Region filtern.

Abrufen einer Liste von aktiven und abgeschlossenen Aufträgen

Im folgenden AWS CLI-Beispiel wird eine Liste von Active und Complete Aufträgen abgerufen.

```
aws s3control list-jobs \  
  --region us-west-2 \  
  --account-id acct-id \  
  --job-statuses '["Active","Complete"]' \  
  --max-results 20
```

Weitere Informationen und Beispiele finden Sie unter [list-jobs](#) in der AWS CLI-Befehlsreferenz.

Anzeigen von Auftragsdetails

Wenn Sie mehr Informationen zu einem Auftrag benötigen, als Sie beim Auflisten von Aufträgen erhalten, können Sie alle Details zu einem einzelnen Auftrag anzeigen lassen. Sie können Details zu Aufträgen anzeigen, die noch nicht abgeschlossen sind, oder zu Aufträgen, die in den letzten 90 Tagen abgeschlossen wurden. Neben den Angaben, die in einer Auftragsliste enthalten sind, umfassen die Details zu einem einzelnen Auftrag auch noch andere Informationen, wie z. B.:

- Die Betriebsparameter
- Details zum Manifest
- Informationen zum Abschlussbericht (falls Sie einen konfiguriert haben, als Sie den Auftrag erstellt haben)

- Der Amazon-Ressourcenname (ARN) der Benutzerrolle, die Sie zum Ausführen des Auftrags zugewiesen haben

Durch das Anzeigen der Details eines einzelnen Auftrags erhalten Sie Zugriff auf die gesamte Konfiguration eines Auftrags. Sie können die Details zu einem Auftrag über die Amazon-S3-Konsole oder die AWS Command Line Interface (AWS CLI) anzeigen.

Die Beschreibung eines Auftrags in S3 Batch Operations in der Amazon-S3-Konsole abrufen

So zeigen Sie die Beschreibung eines Batch-Operations-Auftrags über die Konsole an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie die Auftrags-ID des spezifischen Auftrags aus, um die zugehörigen Details anzuzeigen.

Die Beschreibung eines Auftrags in S3 Batch Operations in der AWS CLI abrufen

Im folgenden Beispiel wird die Beschreibung eines Auftrags in S3 Batch Operations unter Verwendung der AWS CLI abgerufen. Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
aws s3control describe-job \  
--region us-west-2 \  
--account-id acct-id \  
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Weitere Informationen und Beispiele finden Sie unter [describe-job](#) in der AWS CLI-Befehlsreferenz.

Zuweisen der Auftragspriorität

Sie können jedem Auftrag eine numerische Priorität zuweisen, bei der es sich um eine beliebige positive Ganzzahl handeln kann. S3-Batch-Vorgänge priorisiert Aufträge entsprechend der zugewiesenen Priorität. Aufträge mit höherer Priorität (oder einem höheren Wert für den Prioritätsparameter) werden zuerst bewertet. Die Priorität wird in absteigender Reihenfolge ermittelt.

So erhält beispielsweise eine Auftragswarteschlange mit einem Prioritätswert von 10 Vorrang vor einer Auftragswarteschlange mit dem Wert 1.

Sie können die Priorität eines Auftrags auch während seiner Ausführung ändern. Wenn Sie einen neuen Auftrag übermitteln, während ein Auftrag ausgeführt wird, kann der Auftrag mit niedrigerer Priorität pausieren und der Auftrag mit höherer Priorität ausgeführt werden.

Das Ändern der Auftragspriorität hat keinen Einfluss auf die Geschwindigkeit der Auftragsverarbeitung.

Note

S3-Batch-Vorgänge berücksichtigt Auftragsprioritäten auf Best Effort-Basis. Zwar werden Aufträge mit höheren Prioritäten im Allgemeinen vor Aufträgen mit niedrigeren Prioritäten ausgeführt, Amazon S3 garantiert jedoch keine strikte Reihenfolge der Aufträge.

Verwenden der S3-Konsole

So aktualisieren Sie die Auftrags-Priorität in der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie den spezifischen Auftrag aus, den Sie verwalten möchten.
4. Wählen Sie Action (Aktion). Wählen Sie in der Dropdown-Liste Update priority (Priorität aktualisieren).

Verwendung der AWS CLI

Im folgenden Beispiel wird die Auftragspriorität mithilfe der aktualisier AWS CLI. Eine höhere Zahl weist auf eine höhere Ausführungspriorität hin.

```
aws s3control update-job-priority \  
  --region us-west-2 \  
  --account-id acct-id \  
  --priority 98 \  
  --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Verwendung der AWS SDK for Java

Im folgenden Beispiel wird die Priorität eines S3-Batch-Vorgangsauftrags unter Verwendung des AWS SDK for Java aktualisiert.

Weitere Informationen zur Auftragspriorität finden Sie unter [Zuweisen der Auftragspriorität](#).

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobPriorityRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobPriority {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.updateJobPriority(new UpdateJobPriorityRequest()
                .withAccountId(accountId)
                .withJobId(jobId)
                .withPriority(98));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client

```

```

        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}

```

Verfolgen von Auftragsstatus- und Abschluss

Mit S3 Batch Operations können Sie den Auftragsstatus anzeigen und aktualisieren, Benachrichtigungen und Protokollierung hinzufügen, Auftragsfehler verfolgen und Abschlussberichte erstellen.

Themen

- [Auftragsstatus](#)
- [Auftragsstatus wird aktualisiert](#)
- [Benachrichtigungen und Protokollierung](#)
- [Nachverfolgen von Auftragsfehlern](#)
- [Abschlussberichte](#)
- [Beispiele: Nachverfolgen eines S3-Batchoperations-Auftrags in Amazon EventBridge über AWS CloudTrail](#)
- [Beispiele: Abschlussberichte zu S3-Batchvorgänge](#)

Auftragsstatus

Nachdem Sie einen Auftrag erstellt und ausgeführt haben, durchläuft dieser eine Reihe von Status. In der folgenden Tabelle werden der Status und die möglichen Übergänge zwischen ihnen beschrieben.

Status	Beschreibung	Übergänge
New	Ein Auftrag beginnt nach der Erstellung mit dem Status New.	Ein Auftrag wechselt automatisch in den Status <code>Preparing</code> , sobald Amazon S3 mit der Verarbeitung des Manifestobjekts beginnt.
<code>Preparing</code>	Amazon S3 verarbeitet das Manifestobjekt und andere	Ein Auftrag wechselt automatisch in den Status

Status	Beschreibung	Übergänge
	<p>Auftragsparameter, um den Auftrag einzurichten und auszuführen.</p>	<p>Ready, sobald Amazon S3 die Verarbeitung des Manifests und anderer Parameter abschließt. Er ist dann bereit, die angegebene Operation an den im Manifest aufgeführten Objekten auszuführen.</p> <p>Wenn der Auftrag vor der Ausführung bestätigt werden muss, wie dies bei der Erstellung eines Auftrags mit der Amazon-S3-Konsole der Fall ist, wechselt der Auftrag von <code>Preparing</code> zu <code>Suspended</code>. Der Auftrag verbleibt im Status <code>Suspended</code>, bis Sie bestätigen, dass Sie ihn ausführen möchten.</p>

Status	Beschreibung	Übergänge
Suspended	<p>Der Auftrag erfordert eine Bestätigung, aber Sie haben noch nicht bestätigt, dass Sie ihn ausführen möchten. Nur Aufträge, die mit der Amazon-S3-Konsole erstellt wurden, benötigen eine Bestätigung. Ein Auftrag, der mit der Konsole erstellt wurde, wechselt nach <code>Suspended</code> sofort zu <code>Preparing</code>.</p> <p>Nachdem Sie die Ausführung des Auftrags bestätigen und der Auftrag in den Status <code>Ready</code> wechselt, kehrt er nicht wieder in den Status <code>Suspended</code> zurück.</p>	<p>Nachdem Sie die Ausführung des Auftrags bestätigen, wechselt der Status zu <code>Ready</code>.</p>
Ready	<p>Amazon S3 ist bereit, die angeforderten Objekt-Vorgänge auszuführen.</p>	<p>Ein Auftrag wechselt automatisch in den Status <code>Active</code>, sobald Amazon S3 mit dessen Ausführung beginnt. Der Zeitraum, den ein Auftrag im Status <code>Ready</code> verbringt, hängt davon ab, ob bereits Aufträge mit einer höheren Priorität ausgeführt werden und wie lange es dauert, diese Aufträge abzuschließen.</p>

Status	Beschreibung	Übergänge
Active	Amazon S3 führt die angeforderte Operation an den im Manifest aufgeführten Objekten aus. Während ein Job läuft <code>Active</code> , können Sie seinen Fortschritt mit der Amazon S3 S3-Konsole oder den <code>DescribeJob</code> Vorgang über die REST-API oder AWS SDKs überwachen. AWS CLI	Ein Auftrag verlässt den Status <code>Active</code> , wenn er nicht länger Vorgänge an Objekten ausführt. Dies kann automatisch erfolgen, wenn ein Auftrag erfolgreich beendet wird oder fehlschlägt. Oder es kann aufgrund von Benutzeraktionen erfolgen, wenn beispielsweise ein Auftrag abgebrochen wird. Der neue Status des Auftrags hängt vom Grund für den Übergang ab.
Pausing	Der Auftrag wechselt von <code>Paused</code> zu einem anderen Status über.	Ein Auftrag wechselt automatisch in den Status <code>Paused</code> , wenn die Stufe <code>Pausing</code> abgeschlossen ist.
Paused	Ein Auftrag wechselt in den Status <code>Paused</code> , wenn ein anderer Auftrag mit einer höheren Priorität abgeschickt wird, während der aktuelle Auftrag ausgeführt wird.	Ein Auftrag mit dem Status <code>Paused</code> kehrt automatisch in den Status <code>Active</code> zurück, nachdem alle Aufträge mit einer höheren Priorität, die die Ausführung des Auftrags blockierten, abgeschlossen, fehlgeschlagen oder gesperrt sind.

Status	Beschreibung	Übergänge
Complete	Der Auftrag hat die Ausführung der angeforderten Operation an allen Objekten abgeschlossen. Die Operation kann bei den einzelnen Objekten entweder erfolgreich abgeschlossen oder fehlgeschlagen sein. Wenn Sie den Auftrag so konfiguriert haben, dass ein Abschlussbericht generiert werden soll, dann ist der Bericht sofort verfügbar, nachdem der Auftrag in den Status gewechselt ist Complete.	Complete ist ein Beendigungsstatus. Sobald ein Auftrag den Status Complete erreicht, geht er in keinen anderen Status über.
Cancelling	Der Auftrag wechselt in den Status Cancelled.	Ein Auftrag wechselt automatisch in den Status Cancelled, wenn die Stufe Cancelling abgeschlossen ist.
Cancelled	Sie haben angefordert, dass der Auftrag abgebrochen wird, und S3-Batch-Vorgänge hat den Auftrag erfolgreich abgebrochen. Der Auftrag sendet keine neuen Anforderungen an Amazon S3.	Cancelled ist ein Beendigungsstatus. Nachdem ein Auftrag den Status Cancelled erreicht, geht er in keinen anderen Status über.
Failing	Der Auftrag wechselt in den Status Failed.	Ein Auftrag wechselt automatisch in den Status Failed, nachdem die Stufe Failing abgeschlossen ist.

Status	Beschreibung	Übergänge
Failed	Der Auftrag ist fehlgeschlagen und wird nicht länger ausgeführt. Weitere Informationen zu Auftragsfehlern finden Sie unter Nachverfolgen von Auftragsfehlern .	Failed ist ein Beendigungsstatus. Nachdem ein Auftrag den Status Failed erreicht, geht er in keinen anderen Status über.

Auftragsstatus wird aktualisiert

Im Folgenden AWS CLI und in den Beispielen für SDK for Java wird der Status eines Batch Operations-Jobs aktualisiert. Weitere Informationen zur Verwendung der S3-Konsole zur Verwaltung von Batch-Vorgänge-Aufträgen finden Sie unter [Verwenden der Amazon-S3-Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge](#).

Mit dem AWS CLI

- Wenn Sie den Parameter `--no-confirmation-required` im vorherigen `create-job`-Beispiel nicht angegeben haben, bleibt der Auftrag so lange ausgesetzt, bis Sie den Auftrag bestätigen, indem Sie dessen Status auf `Ready` setzen. Amazon S3 erlaubt dann die Ausführung des Auftrags.

```
aws s3control update-job-status \  
  --region us-west-2 \  
  --account-id 181572960644 \  
  --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \  
  --requested-job-status 'Ready'
```

- Brechen Sie den Auftrag ab, indem Sie den Auftragsstatus auf `Cancelled` einstellen.

```
aws s3control update-job-status \  
  --region us-west-2 \  
  --account-id 181572960644 \  
  --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \  
  --status-update-reason "No longer needed" \  
  --requested-job-status Cancelled
```

Verwenden des AWS SDK for Java

Im folgenden Beispiel wird der Status eines S3-Batch-Vorgangsauftrags unter Verwendung des aktualisierten AWS SDK for Java.

Weitere Informationen zum Auftragsstatus finden Sie unter [Verfolgen von Auftragsstatus- und Abschluss](#).

Example

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobStatusRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobStatus {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.updateJobStatus(new UpdateJobStatusRequest()
                .withAccountId(accountId)
                .withJobId(jobId)
                .withRequestedJobStatus("Ready"));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        }
    }
}
```

```
    } catch (SdkClientException e) {  
        // Amazon S3 couldn't be contacted for a response, or the client  
        // couldn't parse the response from Amazon S3.  
        e.printStackTrace();  
    }  
}  
}
```

Benachrichtigungen und Protokollierung

Zusätzlich zum Anfordern von Abschlussberichten können Sie die Batch-Vorgängeaktivitäten auch mithilfe von erfassen, überprüfen und prüfe AWS CloudTrail. Da Batch-Vorgänge vorhandene Amazon-S3-APIs für die Durchführung von Aufgaben nutzt, geben diese Aufgaben dieselben Ereignisse aus, die sie ausgeben würden, wenn Sie sie direkt aufgerufen hätten. So können Sie den Fortschritt Ihres Auftrags und aller seiner Aufgaben mit denselben Benachrichtigungs-, Protokollierungs- und Prüftools und -prozessen nachverfolgen und aufzeichnen, die Sie bereits mit Amazon S3 verwenden. Weitere Informationen finden Sie in den Beispielen in den folgenden Abschnitten.

Note

Amazon S3 Batch Operations generiert CloudTrail während der Auftragsausführung sowohl Verwaltungs- als auch Datenereignisse. Das Volumen dieser Ereignisse wird mit der Anzahl der Schlüssel im Manifest jeder Aufgabe skaliert. Einzelheiten finden Sie auf der Seite mit den [CloudTrail Preisen](#). Dort finden Sie Beispiele dafür, wie sich die Preise je nach der Anzahl der Trails, die Sie in Ihrem Konto konfiguriert haben, ändern. Informationen zum Konfigurieren und Protokollieren von Ereignissen, damit sie Ihren Anforderungen entsprechen, finden Sie unter [Create your first trail \(Erstellen Ihres ersten Trails\)](#) im AWS CloudTrail -Benutzerhandbuch.

Weitere Informationen über Amazon-S3-Ereignisse finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Nachverfolgen von Auftragsfehlern

Wenn bei einem S3-Batch-Vorgangsauftrag ein Problem auftritt, das die erfolgreiche Ausführung verhindert, schlägt der Auftrag fehl. Dies ist beispielsweise der Fall, wenn das angegebene Manifest nicht gelesen werden kann. Wenn ein Auftrag fehlschlägt, generiert er einen oder mehrere

Fehlercodes oder Fehlergründe. S3-Batch-Vorgänge speichert die Fehlercodes und die Gründe mit dem Auftrag, damit Sie sie anzeigen können, indem Sie die Details des Auftrags anfordern. Wenn Sie einen Abschlussbericht für den Auftrag angefordert haben, sind die Fehlercodes und -ursachen auch darin enthalten.

Um zu verhindern, dass zu viele Vorgänge fehlschlagen, wendet Amazon S3 einen Schwellenwert für fehlgeschlagene Aufträge auf jeden Batch-Vorgangsauftrag an. Wenn ein Auftrag mindestens 1.000 Aufgaben ausgeführt hat, überwacht Amazon S3 die Aufgabenfehlerrate. Wenn die Fehlerrate (also die Anzahl fehlgeschlagener Aufgaben im Verhältnis zur Gesamtzahl ausgeführter Aufgaben) zu einem beliebigen Zeitpunkt 50 Prozent überschreitet, schlägt der Auftrag fehl. Wenn Ihr Auftrag fehlschlägt, weil er den Schwellenwert für fehlgeschlagene Aufträge überschritten hat, können Sie die Ursache der Fehler identifizieren. Beispielsweise könnte der Fehler darin liegen, dass Sie versehentlich Objekte in das Manifest aufgenommen haben, die im angegebenen Bucket nicht vorhanden sind. Nach dem Beheben der Fehler können Sie den Auftrag erneut übermitteln.

Note

S3-Batch-Vorgänge wird asynchron betrieben, und die Aufträge werden nicht zwangsläufig in der Reihenfolge ausgeführt, in der die Objekte im Manifest aufgeführt sind. Daher können Sie anhand der Reihenfolge im Manifest nicht feststellen, welche Objektaufgaben erfolgreich ausgeführt wurden und welche fehlgeschlagen sind. Stattdessen können Sie den Abschlussbericht des Auftrags (falls Sie einen angefordert haben) oder Ihre AWS CloudTrail Ereignisprotokolle einsehen, um die Ursache der Fehler zu ermitteln.

Abschlussberichte

Wenn Sie einen Auftrag erstellen, können Sie einen Abschlussbericht anfordern. Solange S3-Batch-Vorgänge mindestens eine Aufgabe erfolgreich aufruft, generiert Amazon S3 einen Abschlussbericht, wenn die Ausführung von Aufträgen abgeschlossen wurde, fehlgeschlagen ist oder abgebrochen wurde. Sie können den Abschlussbericht so konfigurieren, dass er alle oder nur fehlgeschlagene Aufgaben enthält.

Der Abschlussbericht umfasst neben der Auftragskonfiguration und dem Auftragsstatus auch Informationen zu jeder Aufgabe, darunter zu Objektschlüssel und -version, Status, Fehlercodes und Beschreibungen von Fehlern. Abschlussberichte bieten eine einfache Möglichkeit, die Ergebnisse von Aufgaben in einem konsolidierten Format ohne zusätzliche Einrichtung anzuzeigen. Abschlussberichte werden mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt.

Ein Beispiel für einen Abschlussbericht finden Sie unter [Beispiele: Abschlussberichte zu S3-Batchvorgänge](#).

Wenn Sie keinen Abschlussbericht konfigurieren, können Sie Ihren Job und seine Aufgaben trotzdem mithilfe von Amazon überwachen CloudTrail und prüfen CloudWatch. Weitere Informationen finden Sie im folgenden Abschnitt.

Themen

- [Beispiele: Nachverfolgen eines S3-Batchoperations-Auftrags in Amazon EventBridge über AWS CloudTrail](#)
- [Beispiele: Abschlussberichte zu S3-Batchvorgänge](#)

Beispiele: Nachverfolgen eines S3-Batchoperations-Auftrags in Amazon EventBridge über AWS CloudTrail

Die Auftragsaktivität von Amazon S3-Batch-Vorgänge wird in Form von Ereignissen in aufgezeichnete AWS CloudTrail. Sie können eine benutzerdefinierte Regel in Amazon EventBridge erstellen und diese Ereignisse an die gewünschte Zielbenachrichtigungsressource senden, z. B. Amazon Simple Notification Service (Amazon SNS).

Note


Amazon EventBridge ist die bevorzugte Methode zum Verwalten Ihrer Ereignisse. Amazon CloudWatch Events und EventBridge liegen der gleiche Service und die gleiche API zugrunde, EventBridge bietet jedoch mehr Funktionen. Änderungen, die Sie in CloudWatch oder EventBridge vornehmen, werden in allen Konsolen angezeigt. Weitere Informationen finden Sie im [Benutzerhandbuch für Amazon EventBridge](#).

Tracking-Beispiele

- [In CloudTrail aufgezeichnete S3 Batchoperationsereignisse](#)
- [EventBridge-Regel zur Nachverfolgung von S3-Batchoperations-Auftragsereignissen](#)

In CloudTrail aufgezeichnete S3 Batchoperationsereignisse

Wenn ein Batchoperations-Auftrag erstellt wird, wird er als JobCreated-Ereignis in CloudTrail aufgezeichnet. Während der Ausführung des Auftrags ändert sich der Status während der Verarbeitung und andere JobStatusChanged-Ereignisse werden in CloudTrail aufgezeichnet. Sie können diese Ereignisse in der [CloudTrail-Konsole](#) anzeigen. Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

 Note

In CloudTrail werden nur status-change-Ereignisse zu S3-Batch-Vorgänge-Aufträgen aufgezeichnet.

Example In CloudTrail aufgezeichnetes S3-Batchoperations-Auftragsabschlussereignis

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "s3.amazonaws.com"
  },
  "eventTime": "2020-02-05T18:25:30Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "JobStatusChanged",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "s3.amazonaws.com",
  "userAgent": "s3.amazonaws.com",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "f907577b-bf3d-4c53-b9ed-8a83a118a554",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123412341234",
  "serviceEventDetails": {
    "jobId": "d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
    "jobArn": "arn:aws:s3:us-west-2:181572960644:job/d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
    "status": "Complete",
    "jobEventId": "b268784cf0a66749f1a05bce259804f5",
    "failureCodes": [],
    "statusChangeReason": []
  }
}
```

```
}
```

EventBridge-Regel zur Nachverfolgung von S3-Batchoperations-Auftragsereignissen

Das folgende Beispiel zeigt, wie Sie eine Regel in Amazon EventBridge erstellen, um von AWS CloudTrail aufgezeichnete S3-Batchoperations-Ereignisse in einem Ziel Ihrer Wahl erfassen.

Dazu erstellen Sie eine Regel, indem Sie alle Schritte unter [Erstellen von EventBridge-Regeln, die auf Ereignisse reagieren](#) ausführen. Fügen Sie gegebenenfalls die folgende benutzerdefinierte S3-Batchoperations-Ereignismuster-Richtlinie ein und wählen Sie einen Zielservice aus.

Benutzerdefinierte S3-Batchoperationen-Ereignismusterrichtlinie

```
{
  "source": [
    "aws.s3"
  ],
  "detail-type": [
    "AWS Service Event via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3.amazonaws.com"
    ],
    "eventName": [
      "JobCreated",
      "JobStatusChanged"
    ]
  }
}
```

Die folgenden Beispiele zeigen zwei Batchoperations-Ereignisse, die von einer EventBridge-Ereignisregel an Amazon Simple Queue Service (Amazon SQS) gesendet wurden. Ein Batchoperations-Auftrag durchläuft viele verschiedene Zustände während der Verarbeitung (New, Preparing, Active usw.), sodass Sie davon ausgehen können, dass Sie mehrere Nachrichten für jeden Auftrag erhalten.

Example JobCreated Beispielereignis

```
{
  "version": "0",
```

```

    "id": "51dc8145-541c-5518-2349-56d7dffdf2d8",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.s3",
    "account": "123456789012",
    "time": "2020-02-27T15:25:49Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
      "eventVersion": "1.05",
      "userIdentity": {
        "accountId": "11112223334444",
        "invokedBy": "s3.amazonaws.com"
      },
      "eventTime": "2020-02-27T15:25:49Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "JobCreated",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "s3.amazonaws.com",
      "userAgent": "s3.amazonaws.com",
      "eventID": "7c38220f-f80b-4239-8b78-2ed867b7d3fa",
      "readOnly": false,
      "eventType": "AwsServiceEvent",
      "serviceEventDetails": {
        "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
        "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
        "status": "New",
        "jobEventId": "f177ff24f1f097b69768e327038f30ac",
        "failureCodes": [],
        "statusChangeReason": []
      }
    }
  }
}

```

Example JobStatusChanged Auftragsabschluss-Ereignis

```

{
  "version": "0",
  "id": "c8791abf-2af8-c754-0435-fd869ce25233",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.s3",
  "account": "123456789012",
  "time": "2020-02-27T15:26:42Z",

```

```
"region": "us-east-1",
"resources": [],
"detail": {
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "1111222233334444",
    "invokedBy": "s3.amazonaws.com"
  },
  "eventTime": "2020-02-27T15:26:42Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "JobStatusChanged",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "s3.amazonaws.com",
  "userAgent": "s3.amazonaws.com",
  "eventID": "0238c1f7-c2b0-440b-8dbd-1ed5e5833afb",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "serviceEventDetails": {
    "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
    "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
    "status": "Complete",
    "jobEventId": "51f5ac17dba408301d56cd1b2c8d1e9e",
    "failureCodes": [],
    "statusChangeReason": []
  }
}
}
```

Beispiele: Abschlussberichte zu S3-Batchvorgänge

Wenn Sie einen S3-Batchoperations-Auftrag erstellen, können Sie einen Abschlussbericht für alle oder nur für die fehlgeschlagenen Aufträge anfordern. Sofern mindestens eine Aufgabe erfolgreich aufgerufen wurde, generiert S3-Batch-Vorgänge einen Bericht für abgeschlossene, fehlgeschlagene oder abgebrochene Aufträge.

Der Abschlussbericht enthält zusätzliche Informationen zu jeder Aufgabe, darunter den Namen und die Version des Objektschlüssels, Status, Fehlercodes sowie Beschreibungen zu etwaigen Fehlern. Die Beschreibung der Fehler für jede fehlgeschlagene Aufgabe kann herangezogen werden, um Probleme im Rahmen der Auftragserstellung zu diagnostizieren, beispielsweise mit Berechtigungen.

Note

Abschlussberichte werden immer mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt.

Example Manifest-Ergebnisdatei der höchsten Ebene

Die `manifest.json`-Datei der höchsten Ebene enthält die Position jedes Erfolgsberichts und (sofern Fehler in der Auftragsabwicklung aufgetreten sind) jedes Fehlerberichts (siehe folgendes Beispiel).

```
{
  "Format": "Report_CSV_20180820",
  "ReportCreationDate": "2019-04-05T17:48:39.725Z",
  "Results": [
    {
      "TaskExecutionStatus": "succeeded",
      "Bucket": "my-job-reports",
      "MD5Checksum": "83b1c4cbe93fc893f54053697e10fd6e",
      "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/
results/6217b0fab0de85c408b4be96aeaca9b195a7daa5.csv"
    },
    {
      "TaskExecutionStatus": "failed",
      "Bucket": "my-job-reports",
      "MD5Checksum": "22ee037f3515975f7719699e5c416eaa",
      "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/
b2ddad417e94331e9f37b44f1faf8c7ed5873f2e.csv"
    }
  ],
  "ReportSchema": "Bucket, Key, VersionId, TaskStatus, ErrorCode, HTTPStatusCode,
ResultMessage"
}
```

Example Berichte zu fehlgeschlagenen Aufgaben

Berichte zu fehlgeschlagenen Aufgaben enthalten die folgenden Informationen für alle fehlgeschlagenen Aufgaben:

- Bucket

- Key
- VersionId
- TaskStatus
- ErrorCode
- HTTPStatusCode
- ResultMessage

Der folgende Beispielbericht zeigt einen Fall, in dem bei der AWS Lambda Funktion ein Timeout aufgetreten ist, was dazu führte, dass Fehler den Schwellenwert für Fehler überschritten haben. Deshalb wurde die Kennzeichnung zugewiesen `PermanentFailure`.

```
awsexamplebucket1,image_14975,,failed,200,PermanentFailure,"Lambda returned function error: {\"errorMessage\": \"2019-04-05T17:35:21.155Z 2845ca0d-38d9-4c4b-abcf-379dc749c452 Task timed out after 3.00 seconds\"}"
awsexamplebucket1,image_15897,,failed,200,PermanentFailure,"Lambda returned function error: {\"errorMessage\": \"2019-04-05T17:35:29.610Z 2d0a330b-de9b-425f-b511-29232fde5fe4 Task timed out after 3.00 seconds\"}"
awsexamplebucket1,image_14819,,failed,200,PermanentFailure,"Lambda returned function error: {\"errorMessage\": \"2019-04-05T17:35:22.362Z fcf5efde-74d4-4e6d-b37a-c7f18827f551 Task timed out after 3.00 seconds\"}"
awsexamplebucket1,image_15930,,failed,200,PermanentFailure,"Lambda returned function error: {\"errorMessage\": \"2019-04-05T17:35:29.809Z 3dd5b57c-4a4a-48aa-8a35-cbf027b7957e Task timed out after 3.00 seconds\"}"
awsexamplebucket1,image_17644,,failed,200,PermanentFailure,"Lambda returned function error: {\"errorMessage\": \"2019-04-05T17:35:46.025Z 10a764e4-2b26-4d8c-9056-1e1072b4723f Task timed out after 3.00 seconds\"}"
awsexamplebucket1,image_17398,,failed,200,PermanentFailure,"Lambda returned function error: {\"errorMessage\": \"2019-04-05T17:35:44.661Z 1e306352-4c54-4eba-ae8-4d02f8c0235c Task timed out after 3.00 seconds\"}"
```

Example Bericht zu erfolgreichen Aufgaben

Berichte über erfolgreiche Aufgaben enthalten die folgenden Informationen zu den erfolgreichen Aufgaben:

- Bucket
- Key

- VersionId
- TaskStatus
- ErrorCode
- HTTPStatusCode
- ResultMessage

Im folgenden Beispiel hat die Lambda-Funktion das Amazon S3-Objekt in einen anderen Bucket kopiert. Die zurückgegebene Amazon S3-Antwort wird an S3-Batch-Vorgänge zurückgegeben und dann in den endgültigen Abschlussbericht geschrieben.

```
awsexamplebucket1,image_17775,,succeeded,200,, "{u'CopySourceVersionId':
'xVR78haVKlRnurYofbTfYr3ufYbktF8h', u'CopyObjectResult': {u'LastModified':
datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()), u'ETag':
'""fe66f4390c50f29798f040d7aae72784""}, 'ResponseMetadata': {'HTTPStatusCode':
200, 'RetryAttempts': 0, 'HostId': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn00GoXWZpuV0FS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'RequestId': '3ED5852152014362', 'HTTPHeaders':
{'content-length': '234', 'x-amz-id-2': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn00GoXWZpuV0FS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'x-amz-copy-source-version-id':
'xVR78haVKlRnurYofbTfYr3ufYbktF8h', 'server': 'AmazonS3', 'x-amz-request-id':
'3ED5852152014362', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT', 'content-type':
'application/xml'}}}"
awsexamplebucket1,image_17763,,succeeded,200,, "{u'CopySourceVersionId':
'6Hj0USim4Wj6BTcbxToXW44pSZ.40pwq', u'CopyObjectResult': {u'LastModified':
datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()),
u'ETag': '""fe66f4390c50f29798f040d7aae72784""}, 'ResponseMetadata':
{'HTTPStatusCode': 200, 'RetryAttempts': 0, 'HostId': 'GiCZNYr8LHd/
Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'RequestId':
'1BC9F5B1B95D7000', 'HTTPHeaders': {'content-length': '234', 'x-amz-id-2':
'GiCZNYr8LHd/Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'x-
amz-copy-source-version-id': '6Hj0USim4Wj6BTcbxToXW44pSZ.40pwq', 'server': 'AmazonS3',
'x-amz-request-id': '1BC9F5B1B95D7000', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT',
'content-type': 'application/xml'}}}"
awsexamplebucket1,image_17860,,succeeded,200,, "{u'CopySourceVersionId':
'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', u'CopyObjectResult': {u'LastModified':
datetime.datetime(2019, 4, 5, 17, 35, 40, tzinfo=tzlocal()), u'ETag':
'""fe66f4390c50f29798f040d7aae72784""}, 'ResponseMetadata': {'HTTPStatusCode':
200, 'RetryAttempts': 0, 'HostId': 'F9ooZ0gpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'RequestId': '8D9CA56A56813DF3', 'HTTPHeaders':
{'content-length': '234', 'x-amz-id-2': 'F9ooZ0gpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'x-amz-copy-source-version-id':
'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', 'server': 'AmazonS3', 'x-amz-request-id':
```

```
'8D9CA56A56813DF3', 'date': 'Fri, 05 Apr 2019 17:35:40 GMT', 'content-type':  
'application/xml'}}}'
```

Steuern von Zugriffs- und Labeling-Aufträgen mithilfe von Markierungen

Sie können den Zugriff auf Ihre S3-Batch-Vorgänge-Aufträge kennzeichnen und steuern, indem Sie Markierungen hinzufügen. Markierungen können verwendet werden, um zu ermitteln, wer für einen Batchoperations-Auftrag verantwortlich ist. Das Vorhandensein von Aufgabe-Markierungen kann einem Benutzer die Möglichkeit gewähren oder einschränken, eine Aufgabe abzubrechen, eine Aufgabe im Bestätigungsstatus zu aktivieren oder die Prioritätsstufe einer Aufgabe zu ändern. Sie können Aufträge mit ihnen zugeordneten Markierungen erstellen oder Sie können Aufträgen nach ihrer Erstellung Markierungen hinzufügen. Jedes Tag ist ein Schlüssel-Wert-Paar, das beim Erstellen des Auftrags oder später aktualisiert werden kann.

Warning

Auftrags-Markierungen sollten keine vertraulichen Informationen oder persönlichen Daten enthalten.

Betrachten Sie das folgende Tagging-Beispiel: Angenommen, Ihre Finanzabteilung soll einen Batchoperations-Auftrag erstellen. Sie können eine AWS Identity and Access Management (IAM) Richtlinie schreiben, die es einem Benutzer ermöglicht `CreateJob` aufzurufen, vorausgesetzt, der Auftrag wird mit dem `Department`-Tag erstellt, der den zugewiesenen Wert `Finance` hat. Außerdem können Sie diese Richtlinie an alle Benutzer anfügen, die Mitglieder der Finanzabteilung sind.

Wenn Sie mit diesem Beispiel fortfahren, können Sie eine Richtlinie schreiben, die es einem Benutzer ermöglicht, die Priorität aller Aufträge, die über die gewünschten Markierungen verfügen, zu aktualisieren oder alle Aufträge, die diese Markierungen besitzen, abzubrechen. Weitere Informationen finden Sie unter [the section called "Steuern von Berechtigungen"](#).

Sie können Markierungen beim Erstellen neuen S3-Batchoperations-Aufträgen oder bereits vorhandenen Aufträgen hinzufügen.

Beachten Sie die folgenden Tag-Einschränkungen:

- Sie können einem Auftrag bis zu 50 Markierungen zuordnen, solange diese über eindeutige Tag-Schlüssel verfügen.

- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

Weitere Informationen zu Tag-Einschränkungen finden Sie unter [Einschränkungen benutzerdefinierter Markierungen](#) im AWS Billing and Cost Management Benutzerhandbuch.

API-Vorgänge im Zusammenhang mit S3-Batchoperations-Auftragsmarkierung

Amazon S3 unterstützt die folgenden API-Vorgänge, die spezifisch für das S3-Batchoperations-Auftrag-Tagging sind:

- [GetJobTagging](#) – Gibt den Markierungenatz zurück, der mit einem Batchoperations-Auftrag verknüpft ist.
- [PutJobTagging](#) – Ersetzt den Markierungenatz, der einem Auftrag zugeordnet ist. Es gibt zwei unterschiedliche Szenarien der S3-Batchoperations-Auftrags-Tag-Verwaltung unter Verwendung dieser API-Aktion:
 - Auftrag ohne Markierungen – Sie können einen Satz von Markierungen zu einem Auftrag hinzufügen (der Auftrag hat keine vorherigen Markierungen).
 - Auftrag verfügt über einen Satz vorhandener Markierungen – Um den vorhandenen Tag-Satz zu ändern, können Sie entweder den vorhandenen Tag-Satz vollständig ersetzen oder Änderungen innerhalb des vorhandenen Tag-Satzes vornehmen, indem Sie den vorhandenen Tag-Satz mit [GetJobTagging](#) abrufen, diesen Tag-Satz ändern und mit dieser API-Aktion diesen Tag-Satz gegen einen von Ihnen modifizierten auswechseln.

Note

Wenn Sie diese Anforderung mit einem leeren Tag-Satz senden, löscht S3-Batch-Vorgänge den vorhandenen Tag-Satz für das Objekt. Wenn Sie diese Methode verwenden, wird Ihnen eine Tier 1-Anforderung (PUT) berechnet. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Wenn Sie vorhandene Markierungen für Ihren Batchoperations-Auftrag löschen möchten, wird die Aktion `DeleteJobTagging` bevorzugt, da sie dasselbe Ergebnis ohne anfallende Gebühren erzielt.

- [DeleteJobTagging](#) – Löscht den Markierungenatz, der mit einem Batchoperations-Auftrag verknüpft ist.

Erstellen eines Batchoperations-Auftrags mit Auftrags-Markierungen zur Kennzeichnung

Sie können den Zugriff auf Ihre S3-Batch-Vorgänge-Aufträge kennzeichnen und steuern, indem Sie Markierungen hinzufügen. Markierungen können verwendet werden, um zu ermitteln, wer für einen Batchoperations-Auftrag verantwortlich ist. Sie können Aufträge mit ihnen zugeordneten Markierungen erstellen oder Sie können Aufträgen nach ihrer Erstellung Markierungen hinzufügen. Weitere Informationen finden Sie unter [the section called "Verwenden von Markierungen"](#).

Verwendung der AWS CLI

Im folgenden AWS CLI-CLI-Beispiel wird der S3-Batchoperations-Auftrag `S3PutObjectCopy` mit Auftrags-Markierungen als Bezeichnungen für den Auftrag erstellt.

1. Wählen Sie die Aktion oder `OPERATION` aus, die der Batchoperations-Auftrag ausführen soll, und wählen Sie Ihre `TargetResource`.

```
read -d '' OPERATION <<EOF
{
  "S3PutObjectCopy": {
    "TargetResource": "arn:aws:s3:::destination-bucket"
  }
}
EOF
```

2. Identifizieren Sie die Auftrags-TAGS, die Sie für den Auftrag wünschen. In diesem Fall wenden Sie beiden Markierungen `department` und `FiscalYear` mit den Werten `Marketing` bzw. `2020` an.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

3. Geben Sie das MANIFEST für den Batchoperations-Auftrag an.

```
read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "EXAMPLE_S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::example-bucket/example_manifest.csv",
    "ETag": "example-5dc7a8bf90808fc5d546218"
  }
}
EOF
```

4. Konfigurieren Sie den REPORT für den Batchoperations-Auftrag.

```
read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::example-report-bucket",
  "Format": "Example_Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/copy-with-replace-metadata",
  "ReportScope": "AllTasks"
}
EOF
```

5. Führen Sie die Aktion `create-job` aus, um Ihren Batchoperations-Auftrag mit Eingaben zu erstellen, die in den vorherigen Schritten festgelegt wurden.

```
aws \
  s3control create-job \
  --account-id 123456789012 \
  --manifest "${MANIFEST//$\n}" \
  --operation "${OPERATION//$\n/}" \
  --report "${REPORT//$\n}" \
  --priority 10 \
  --role-arn arn:aws:iam::123456789012:role/batch-operations-role \
  --tags "${TAGS//$\n/}" \
  --client-request-token "$(uuidgen)" \
```

```
--region us-west-2 \  
--description "Copy with Replace Metadata";
```

Verwenden des AWS-SDK für Java

Example

Im folgenden Beispiel wird ein S3-Batchoperations-Auftrag mit Markierungen unter Verwendung des erstellten AWS SDK for Java.

```
public String createJob(final AWSS3ControlClient awss3ControlClient) {  
    final String manifestObjectArn = "arn:aws:s3:::example-manifest-bucket/  
manifests/10_manifest.csv";  
    final String manifestObjectVersionId = "example-5dc7a8bfb90808fc5d546218";  
  
    final JobManifestLocation manifestLocation = new JobManifestLocation()  
        .withObjectArn(manifestObjectArn)  
        .withETag(manifestObjectVersionId);  
  
    final JobManifestSpec manifestSpec =  
        new  
        JobManifestSpec().withFormat(JobManifestFormat.S3InventoryReport_CSV_20161130);  
  
    final JobManifest manifestToPublicApi = new JobManifest()  
        .withLocation(manifestLocation)  
        .withSpec(manifestSpec);  
  
    final String jobReportBucketArn = "arn:aws:s3:::example-report-bucket";  
    final String jobReportPrefix = "example-job-reports";  
  
    final JobReport jobReport = new JobReport()  
        .withEnabled(true)  
        .withReportScope(JobReportScope.AllTasks)  
        .withBucket(jobReportBucketArn)  
        .withPrefix(jobReportPrefix)  
        .withFormat(JobReportFormat.Report_CSV_20180820);  
  
    final String lambdaFunctionArn = "arn:aws:lambda:us-  
west-2:123456789012:function:example-function";  
  
    final JobOperation jobOperation = new JobOperation()  
        .withLambdaInvoke(new  
        LambdaInvokeOperation().withFunctionArn(lambdaFunctionArn));
```

```
final S3Tag departmentTag = new
S3Tag().withKey("department").withValue("Marketing");
final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

final String roleArn = "arn:aws:iam::123456789012:role/example-batch-operations-
role";
final Boolean requiresConfirmation = true;
final int priority = 10;

final CreateJobRequest request = new CreateJobRequest()
    .withAccountId("123456789012")
    .withDescription("Test lambda job")
    .withManifest(manifestToPublicApi)
    .withOperation(jobOperation)
    .withPriority(priority)
    .withRoleArn(roleArn)
    .withReport(jobReport)
    .withTags(departmentTag, fiscalYearTag)
    .withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.getJobId();
}
```

Löschen der Markierungen aus einem S3-Batchoperations-Auftrag

Sie können diese Beispiele verwenden, um die Markierungen aus einem Batchoperations-Auftrag zu löschen.

Verwendung der AWS CLI

Im folgenden Beispiel werden die Markierungen aus einem Batchoperations-Auftrag unter Verwendung der gelöschte AWS CLI.

```
aws \
  s3control delete-job-tagging \
  --account-id 123456789012 \
  --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
  --region us-east-1;
```

Löschen der Auftrags-Markierungen eines Batchoperations-Auftrags

Example

Im folgenden Beispiel werden die Markierungen eines S3-Batchoperations-Auftrags unter Verwendung des -SDK für Java gelöscht AWS SDK for Java.

```
public void deleteJobTagging(final AWSS3ControlClient awss3ControlClient,
                            final String jobId) {
    final DeleteJobTaggingRequest deleteJobTaggingRequest = new
DeleteJobTaggingRequest()
        .withJobId(jobId);

    final DeleteJobTaggingResult deleteJobTaggingResult =
        awss3ControlClient.deleteJobTagging(deleteJobTaggingRequest);
}
```

Einfügen von Auftrags-Markierungen für einen bestehenden S3-Batchoperations-Auftrag

Sie können [PutJobTagging](#) verwenden, um Auftrags-Markierungen zu Ihren bestehenden S3-Batchoperationsaufträgen hinzuzufügen. Weitere Informationen finden Sie in den folgenden Beispielen.

Verwendung der AWS CLI

Im Folgenden finden Sie ein Beispiel für `s3control put-job-tagging` das Hinzufügen von Auftrags-Markierungen zu Ihrem S3-Batchoperations-Auftrag unter der Verwendung der AWS CLI.

Note

Wenn Sie diese Anforderung mit einem leeren Tag-Satz senden, löscht S3-Batch-Vorgänge den vorhandenen Tag-Satz für das Objekt. Wenn Sie diese Methode verwenden, wird Ihnen eine Tier 1-Anforderung (PUT) berechnet. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Wenn Sie vorhandene Markierungen für Ihren Batchoperations-Auftrag löschen möchten, wird die Aktion `DeleteJobTagging` bevorzugt, da sie dasselbe Ergebnis ohne anfallende Gebühren erzielt.

1. Identifizieren Sie die Auftrags-TAGS, die Sie für den Auftrag wünschen. In diesem Fall wenden Sie beiden Markierungen `department` und `FiscalYear` mit den Werten `Marketing` bzw. `2020` an.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

2. Führen Sie die `put-job-tagging`-Aktion mit den erforderlichen Parametern aus.

```
aws \
  s3control put-job-tagging \
  --account-id 123456789012 \
  --tags "${TAGS//$\n'/}" \
  --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
  --region us-east-1;
```

Verwenden des AWS-SDK für Java

Example

Im folgenden Beispiel werden die Markierungen eines S3-Batchoperations-Auftrags unter Verwendung des -SDK für Java eingefügt AWS SDK for Java.

```
public void putJobTagging(final AWSS3ControlClient awss3ControlClient,
                          final String jobId) {
    final S3Tag departmentTag = new
    S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final PutJobTaggingRequest putJobTaggingRequest = new PutJobTaggingRequest()
        .withJobId(jobId)
```

```
        .withTags(departmentTag, fiscalYearTag);

    final PutJobTaggingResult putJobTaggingResult =
    awss3ControlClient.putJobTagging(putJobTaggingRequest);
}
```

Abrufen der Markierungen eines S3-Batchoperations-Auftrags

Sie können `GetJobTagging` verwenden, um die Markierungen eines S3-Batchoperations-Auftrags zurückzugeben. Weitere Informationen finden Sie in den folgenden Beispielen.

Verwendung der AWS CLI

Im folgenden Beispiel werden die Markierungen eines Batchoperations-Auftrags unter Verwendung der CLI abgerufen AWS CLI.

```
aws \
  s3control get-job-tagging \
  --account-id 123456789012 \
  --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
  --region us-east-1;
```

Verwenden des AWS-SDK für Java

Example

Im folgenden Beispiel werden die Markierungen eines S3 Batchoperations-Auftrags unter Verwendung der CLI abgerufen AWS SDK for Java.

```
public List<S3Tag> getJobTagging(final AWSS3ControlClient awss3ControlClient,
                                final String jobId) {
    final GetJobTaggingRequest getJobTaggingRequest = new GetJobTaggingRequest()
        .withJobId(jobId);

    final GetJobTaggingResult getJobTaggingResult =
        awss3ControlClient.getJobTagging(getJobTaggingRequest);

    final List<S3Tag> tags = getJobTaggingResult.getTags();

    return tags;
}
```


Steuern von Berechtigungen für S3-Batch-Vorgänge mithilfe von Auftrags-Markierungen

Zur Unterstützung bei der Verwaltung Ihrer S3-Batch-Vorgänge-Aufträge können Sie Auftrags-Markierungen hinzufügen. Mit Auftrags-Markierungen können Sie den Zugriff auf Ihre BatchVorgängeaufträge steuern und erzwingen, dass Markierungen angewendet werden, wenn ein Auftrag erstellt wird.

Sie können bis zu 50 Auftrag-Markierungen auf jeden Batchoperations-Auftrag anwenden. Auf diese Weise können Sie sehr detaillierte Richtlinien festlegen, die die Gruppe von Benutzern einschränken, die den Auftrag bearbeiten können. Aufgaben-Markierungen können einem Benutzer die Möglichkeit gewähren oder einschränken, eine Aufgabe abzubrechen, eine Aufgabe im Bestätigungsstatus zu aktivieren oder die Prioritätsstufe einer Aufgabe zu ändern. Darüber hinaus können Sie erzwingen, dass Markierungen auf alle neuen Aufträge angewendet werden, und die zulässigen Schlüssel-Wert-Paare für die Markierungen angeben. Sie können all diese Bedingungen in derselben [IAMRichtliniensprache](#) ausdrücken. Weitere Informationen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3](#) in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).

Das folgende Beispiel zeigt, wie Sie S3-Batchoperations-Auftrags-Markierungen verwenden können, um Benutzern die Berechtigung zu erteilen, nur die Aufträge zu erstellen und zu bearbeiten, die in einer bestimmten Abteilung (z. B. der Abteilung Finanzen oder Compliance) ausgeführt werden. Sie können Aufträge auch basierend auf der Entwicklungsphase zuweisen, auf die sie sich beziehen, z. B. QS oder Produktion.

In diesem Beispiel verwenden Sie die Job-Tags von S3 Batch Operations in AWS Identity and Access Management (IAM) Richtlinien, mit denen Benutzern nur die Erlaubnis erteilt wird, Jobs zu erstellen und zu bearbeiten, die in ihrer Abteilung ausgeführt werden. Sie weisen Aufträge basierend auf der Entwicklungsphase zu, mit der sie verbunden sind, wie QS oder Produktion.

In diesem Beispiel werden die folgenden Abteilungen verwendet, die Batch-Vorgänge auf jeweils unterschiedliche Weise einsetzen:

- Finanzen
- Compliance
- Business Intelligence

- Entwicklung

Themen

- [Steuern des Zugriffs durch Zuweisen von Markierungen zu Benutzern und Ressourcen](#)
- [Markieren von BatchVorgängeaufträgen nach Stufe und Durchsetzen von Limits für die Auftragspriorität](#)

Steuern des Zugriffs durch Zuweisen von Markierungen zu Benutzern und Ressourcen

In diesem Szenario verwenden die Administratoren die [attributbasierte Zugriffskontrolle](#) (). ABAC ist eine IAM Autorisierungsstrategie, bei der Berechtigungen definiert werden, indem sowohl Benutzern als auch Benutzern Tags zugewiesen werden AWS Ressourcen schützen.

Benutzern und Jobs wird eines der folgenden Abteilungstags zugewiesen:

Schlüssel : Wert

- department : Finance
- department : Compliance
- department : BusinessIntelligence
- department : Engineering

Note

Bei Tag-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden.

Mithilfe der ABAC Zugriffskontrollstrategie gewähren Sie einem Benutzer in der Finanzabteilung die Berechtigung, S3 Batch Operations-Jobs in seiner Abteilung zu erstellen und zu verwalten, indem Sie das Tag department=Finance seinem Benutzer zuordnen.

Darüber hinaus können Sie dem IAM Benutzer eine verwaltete Richtlinie zuordnen, die es jedem Benutzer in seinem Unternehmen ermöglicht, S3 Batch Operations-Jobs in seinen jeweiligen Abteilungen zu erstellen oder zu ändern.

Die Richtlinie in diesem Beispiel enthält drei Richtlinienanweisungen:

- Die erste Anweisung in der Richtlinie ermöglicht es dem Benutzer, einen Batchoperations-Auftrag zu erstellen, vorausgesetzt, die Auftragserstellungsanforderung enthält ein Auftrags-Tag, das der jeweiligen Abteilung entspricht. Dies wird mithilfe der Syntax "\${aws:PrincipalTag/department}" ausgedrückt, die zum Zeitpunkt der Richtlinienauswertung durch das Abteilungs-Tag des Benutzers ersetzt wird. Die Bedingung ist erfüllt, wenn der für das Abteilungs-Tag in der Anforderung ("aws:RequestTag/department") angegebene Wert mit dem der Abteilung des Benutzers übereinstimmt.
- Die zweite Anweisung in der Richtlinie ermöglicht es Benutzern, die Priorität von Aufträgen zu ändern oder den Status eines Auftrags zu aktualisieren, sofern der Auftrag, den der Benutzer aktualisiert, mit der Abteilung des Benutzers übereinstimmt.
- Die dritte Anweisung ermöglicht es einem Benutzer, die Markierungen eines Batchoperations-Auftrags jederzeit über eine PutJobTagging-Anforderung zu aktualisieren, solange (1) das Abteilungs-Tag erhalten bleibt und (2) sich der Auftrag, der aktualisiert wird, innerhalb der Abteilung des Benutzers befindet.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:CreateJob",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/department": "${aws:PrincipalTag/department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:UpdateJobPriority",
        "s3:UpdateJobStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```

    "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
    }
  },
  {
    "Effect": "Allow",
    "Action": "s3:PutJobTagging",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
        "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
      }
    }
  }
]
}

```

Markieren von BatchVorgängeaufträgen nach Stufe und Durchsetzen von Limits für die Auftragspriorität

Alle S3-BatchVorgängeaufträge haben eine numerische Priorität, anhand derer Amazon S3 entscheidet, in welcher Reihenfolge die Aufträge ausgeführt werden sollen. In diesem Beispiel beschränken Sie die maximale Priorität, die die meisten Benutzer Aufträgen zuweisen können, wobei höhere Prioritätsbereiche für eine begrenzte Gruppe von berechtigten Benutzern reserviert sind:

- Prioritätsbereich der QA-Stufe (niedrig): 1-100
- Prioritätsbereich der Produktionsstufe (hoch): 1-300

Dazu führen Sie einen neuen Tag-Satz ein, der die Phase des Auftrags repräsentiert:

Schlüssel : Wert

- stage : QA
- stage : Production

Erstellen und Aktualisieren von Aufträgen mit geringer Priorität in einer Abteilung

Mit dieser Richtlinie werden neben der abteilungsbasierten Einschränkung zwei neue Einschränkungen für die Erstellung und Aktualisierung von S3-BatchVorgängeaufträgen eingeführt:

- Es ermöglicht Benutzern, Aufträge in ihrer Abteilung mit einer neuen Bedingung zu erstellen oder zu aktualisieren, die erfordert, dass der Auftrag das Tag enthält `stage=QA`.
- Es ermöglicht Benutzern, Aufträge mit einer maximalen Priorität von bis 100 zu erstellen oder zu aktualisieren.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:CreateJob",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/department": "${aws:PrincipalTag/department}",
          "aws:RequestTag/stage": "QA"
        },
        "NumericLessThanEquals": {
          "s3:RequestJobPriority": 100
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:UpdateJobStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
        }
      }
    },
    {
      "Effect": "Allow",
```

```

    "Action": "s3:UpdateJobPriority",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
        "aws:ResourceTag/stage": "QA"
      },
      "NumericLessThanEquals": {
        "s3:RequestJobPriority": 100
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "s3:PutJobTagging",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/department" : "${aws:PrincipalTag/department}",
        "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
        "aws:RequestTag/stage": "QA",
        "aws:ResourceTag/stage": "QA"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "s3:GetJobTagging",
    "Resource": "*"
  }
]
}

```

Erstellen und Aktualisieren von Aufträgen mit hoher Priorität in einer Abteilung

Einer kleinen Anzahl von Benutzern muss u. U. möglich sein, Aufträge hoher Priorität in QS oder Produktion zu erstellen. Um diesen Bedarf zu unterstützen, erstellen Sie eine verwaltete Richtlinie durch Abwandlung einer Richtlinie mit niedriger Priorität im vorherigen Abschnitt.

Diese Richtlinie gewährt die folgenden Aktionen:

- Ermöglicht Benutzern, Aufträge in ihrer Abteilung entweder mit dem Tag `stage=QA` oder `stage=Production` zu erstellen oder zu aktualisieren.

- Ermöglicht Benutzern, bei der Erstellung oder Aktualisierung eines Auftrags als Priorität bis zu 300 anzugeben.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:CreateJob",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:RequestTag/stage": [
            "QA",
            "Production"
          ]
        },
        "StringEquals": {
          "aws:RequestTag/department": "${aws:PrincipalTag/department}"
        },
        "NumericLessThanEquals": {
          "s3:RequestJobPriority": 300
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:UpdateJobStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "s3:UpdateJobPriority",
```

```

        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "aws:ResourceTag/stage": [
                    "QA",
                    "Production"
                ]
            },
            "StringEquals": {
                "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
            },
            "NumericLessThanEquals": {
                "s3:RequestJobPriority": 300
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:PutJobTagging",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
            },
            "ForAnyValue:StringEquals": {
                "aws:RequestTag/stage": [
                    "QA",
                    "Production"
                ],
                "aws:ResourceTag/stage": [
                    "QA",
                    "Production"
                ]
            }
        }
    }
]
}

```


Verwalten der S3-Objektsperre mit S3-BatchVorgänge

Mit der S3-Objektsperre können Sie eine rechtliche Aufbewahrungsfrist für eine Objektversion festlegen. Wie das Festlegen von Aufbewahrungszeiträumen verhindern auch rechtliche Aufbewahrungsfristen das Überschreiben oder Löschen von Objektversionen. Mit rechtlichen Aufbewahrungsfristen sind jedoch keine Aufbewahrungszeiträume verknüpft. Sie sind gültig, bis sie entfernt werden. Weitere Informationen finden Sie unter [S3-Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen](#).

Informationen zur Verwendung von S3-Batch-Vorgängen mit Object Lock zum Hinzufügen von rechtlichen Holds zu vielen Amazon-S3-Objekten gleichzeitig finden Sie in den folgenden Abschnitten.

Themen

- [Aktivieren der S3-Objektsperre mit S3-BatchVorgänge](#)
- [Festlegen der Objektsperrenaufbewahrung mit BatchVorgänge](#)
- [Verwendung von S3-Batch-Vorgänge mit dem Compliance-Modus der S3-Objektsperrenaufbewahrung](#)
- [Verwenden von S3-Batch-Vorgänge mit dem Governance-Modus der S3-Objektsperrenaufbewahrung](#)
- [Verwendung rechtlicher Aufbewahrungsfristen für die S3-Objektsperre mithilfe von S3-Batch-Vorgänge deaktivieren](#)

Aktivieren der S3-Objektsperre mit S3-BatchVorgänge

Sie können S3-Batch-Vorgänge mit der S3-Objektsperre verwenden, um für viele Amazon-S3-Objekte gleichzeitig die Aufbewahrung zu verwalten oder eine rechtliche Aufbewahrungsfrist zu wählen. Sie geben die Liste der Zielobjekte in Ihrem Manifest an und senden sie zur Fertigstellung an BatchVorgänge. Weitere Informationen erhalten Sie unter [the section called “Aufrechterhaltung der Objektsperre”](#) und [the section called “Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen”](#).

In den folgenden Beispielen wird gezeigt, wie Sie eine IAM-Rolle mit Berechtigungen für S3-Batch-Vorgänge erstellen und die Rollenberechtigungen aktualisieren, um Aufträge zu erstellen, die die Objektsperre aktivieren. Ersetzen Sie in den Beispielen alle Variablenwerte durch diejenigen, die Ihren Anforderungen entsprechen. Sie müssen auch über ein CSV-Manifest verfügen, das die Objekte für Ihren S3-Batch-Vorgangsauftrag identifiziert. Weitere Informationen finden Sie unter [the section called “Angeben eines Manifests”](#).

Mit dem AWS CLI

1. Erstellen Sie eine IAM-Rolle und weisen Sie zur Ausführung Berechtigungen für S3-Batch-Vorgänge zu.

Dieser Schritt ist für alle S3-BatchVorgängeaufträge erforderlich.

```
export AWS_PROFILE='aws-user'

read -d '' bops_trust_policy <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "batchoperations.s3.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
aws iam create-role --role-name bops-objectlock --assume-role-policy-document
"${bops_trust_policy}"
```

2. Richten Sie S3-Batch-Vorgänge zur Ausführung mit S3-Objektsperre ein.

In diesem Schritt lassen Sie der Rolle Folgendes zu:

- a. Führen Sie die Objektsperre in dem S3-Bucket aus, der die Zielobjekte enthält, für die Batch-Vorgänge ausgeführt werden soll.
- b. Lesen des S3-Buckets, in dem sich die Manifest-CSV-Datei und die Objekte befinden.
- c. Schreiben Sie die Ergebnisse des S3-Batch-Vorgangsauftrags in den Berichts-Bucket.

```
read -d '' bops_permissions <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Effect": "Allow",
      "Action": "s3:GetBucketObjectLockConfiguration",
      "Resource": [
        "arn:aws:s3:::{{ManifestBucket}}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::{{ManifestBucket}}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::{{ReportBucket}}/*"
      ]
    }
  ]
}
EOF

```

```
aws iam put-role-policy --role-name bops-objectlock --policy-name object-lock-permissions --policy-document "${bops_permissions}"
```

Verwenden des AWS SDK for Java

In den folgenden Beispielen wird gezeigt, wie Sie eine IAM-Rolle mit Berechtigungen für S3-Batch-Vorgänge erstellen und die Rollenberechtigungen aktualisieren, um Aufträge zu erstellen, die die Objektsperre mithilfe des AWS SDK for Java aktivieren. Ersetzen Sie im Code alle Variablenwerte durch diejenigen, die Ihren Anforderungen entsprechen. Sie müssen auch über ein CSV-Manifest

verfügen, das die Objekte für Ihren S3-Batch-Vorgangsauftrag identifiziert. Weitere Informationen finden Sie unter [the section called "Angeben eines Manifests"](#).

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie eine IAM-Rolle und weisen Sie zur Ausführung Berechtigungen für S3-Batch-Vorgänge zu. Dieser Schritt ist für alle S3-BatchVorgängeaufträge erforderlich.
2. Richten Sie S3-Batch-Vorgänge zur Ausführung mit S3-Objektsperre ein.

Sie lassen der Rolle Folgendes zu:

1. Führen Sie die Objektsperre in dem S3-Bucket aus, der die Zielobjekte enthält, für die Batch-Vorgänge ausgeführt werden soll.
2. Lesen des S3-Buckets, in dem sich die Manifest-CSV-Datei und die Objekte befinden.
3. Schreiben Sie die Ergebnisse des S3-Batch-Vorgangsauftrags in den Berichts-Bucket.

```
public void createObjectLockRole() {
    final String roleName = "bops-object-lock";

    final String trustPolicy = "{" +
        "  \"Version\": \"2012-10-17\", " +
        "  \"Statement\": [ " +
        "    { " +
        "      \"Effect\": \"Allow\", " +
        "      \"Principal\": { " +
        "        \"Service\": [ " +
        "          \"batchoperations.s3.amazonaws.com\" " +
        "        ] " +
        "      }, " +
        "      \"Action\": \"sts:AssumeRole\" " +
        "    } " +
        "  ] " +
        "};

    final String bopsPermissions = "{" +
        "  \"Version\": \"2012-10-17\", " +
        "  \"Statement\": [ " +
        "    { " +
        "      \"Effect\": \"Allow\", " +
        "      \"Action\": \"s3:GetBucketObjectLockConfiguration\", " +
        "      \"Resource\": [ " +
```

```

"          \"arn:aws:s3:::ManifestBucket\" +
"      ]" +
"  }," +
"  {" +
"    \"Effect\": \"Allow\"," +
"    \"Action\": [" +
"      \"s3:GetObject\"," +
"      \"s3:GetObjectVersion\"," +
"      \"s3:GetBucketLocation\" +
"    ]," +
"    \"Resource\": [" +
"      \"arn:aws:s3:::ManifestBucket/*\" +
"    ]" +
"  }," +
"  {" +
"    \"Effect\": \"Allow\"," +
"    \"Action\": [" +
"      \"s3:PutObject\"," +
"      \"s3:GetBucketLocation\" +
"    ]," +
"    \"Resource\": [" +
"      \"arn:aws:s3:::ReportBucket/*\" +
"    ]" +
"  }" +
" ]" +
"}";

```

```

final AmazonIdentityManagement iam =
    AmazonIdentityManagementClientBuilder.defaultClient();

```

```

final CreateRoleRequest createRoleRequest = new CreateRoleRequest()
    .withAssumeRolePolicyDocument(bopsPermissions)
    .withRoleName(roleName);

```

```

final CreateRoleResult createRoleResult = iam.createRole(createRoleRequest);

```

```

final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
    .withPolicyDocument(bopsPermissions)
    .withPolicyName("bops-permissions")
    .withRoleName(roleName);

```

```

final PutRolePolicyResult putRolePolicyResult =
iam.putRolePolicy(putRolePolicyRequest);

```

```
}
```

Festlegen der Objektsperrenaufbewahrung mit Batchvorgänge

Im folgenden Beispiel kann die Regel die S3-Objektsperrenaufbewahrung für Ihre Objekte im Manifest-Bucket festlegen.

Sie aktualisieren die Rolle, sodass sie `s3:PutObjectRetention`-Berechtigungen einschließt und Sie die Objektsperrenaufbewahrung für die Objekte in Ihrem Bucket ausführen können.

Mit dem AWS CLI

```
export AWS_PROFILE='aws-user'

read -d '' retention_permissions <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectRetention"
      ],
      "Resource": [
        "arn:aws:s3:::{{ManifestBucket}}/*"
      ]
    }
  ]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name retention_permissions
--policy-document "${retention_permissions}"
```

Verwenden des AWS SDK for Java

```
public void allowPutObjectRetention() {
    final String roleName = "bops-object-lock";

    final String retentionPermissions = "{" +
        "  \"Version\": \"2012-10-17\", " +
        "  \"Statement\": [" +
        "    {" +
```

```

    "          \"Effect\": \"Allow\",\" +
    "          \"Action\": [\" +
    "              \"s3:PutObjectRetention\" +
    "          ],\" +
    "          \"Resource\": [\" +
    "              \"arn:aws:s3:::ManifestBucket*\" +
    "          ]\" +
    "      }\" +
    "  ]\" +
    "};";

```

```

final AmazonIdentityManagement iam =
    AmazonIdentityManagementClientBuilder.defaultClient();

final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
    .withPolicyDocument(retentionPermissions)
    .withPolicyName("retention-permissions")
    .withRoleName(roleName);

final PutRolePolicyResult putRolePolicyResult =
    iam.putRolePolicy(putRolePolicyRequest);
}

```

Verwendung von S3-Batch-Vorgänge mit dem Compliance-Modus der S3-Objektsperrenaufbewahrung

Das folgende Beispiel baut auf den vorherigen Beispielen zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs und S3-Objektsperren-Konfigurations-Berechtigungen für Ihre Objekte auf. In diesem Beispiel wird der Aufbewahrungsmodus auf COMPLIANCE und dann `retain until date` auf den 1. Januar 2025 festgelegt. Es erstellt einen Auftrag, der auf Objekte im Manifest-Bucket abzielt und die Ergebnisse im Berichts-Bucket meldet, den Sie identifiziert haben.

Unter Verwendung des AWS CLI

Example Festlegen der Compliance von Erwähnungen für mehrere Objekte

```

export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF

```

```

{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate": "2025-01-01T00:00:00",
      "Mode": "COMPLIANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/compliance-objects-bops",
  "ReportScope": "AllTasks"
}
EOF

aws \
  s3control create-job \
  --account-id "${ACCOUNT_ID}" \
  --manifest "${MANIFEST//$'\n'}" \
  --operation "${OPERATION//$'\n'/'}" \
  --report "${REPORT//$'\n'}" \
  --priority 10 \
  --role-arn "${ROLE_ARN}" \

```



```
--client-request-token "$(uuidgen)" \  
--region "${AWS_DEFAULT_REGION}" \  
--description "Set compliance retain-until to 1 Jul 2030";
```

Example Verlängern Sie die **COMPLIANCE** Modi **retain until date** bis zum 15. Januar 2025

Im folgenden Beispiel wird das COMPLIANCE des `retain until date`-Modus bis zum 15. Januar 2025 erweitert.

```
export AWS_PROFILE='aws-user'  
export AWS_DEFAULT_REGION='us-west-2'  
export ACCOUNT_ID=123456789012  
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'  
  
read -d '' OPERATION <<EOF  
{  
  "S3PutObjectRetention": {  
    "Retention": {  
      "RetainUntilDate": "2025-01-15T00:00:00",  
      "Mode": "COMPLIANCE"  
    }  
  }  
}  
EOF  
  
read -d '' MANIFEST <<EOF  
{  
  "Spec": {  
    "Format": "S3BatchOperations_CSV_20180820",  
    "Fields": [  
      "Bucket",  
      "Key"  
    ]  
  },  
  "Location": {  
    "ObjectArn": "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv",  
    "ETag": "Your-manifest-ETag"  
  }  
}  
EOF  
  
read -d '' REPORT <<EOF  
{
```

```

"Bucket": "arn:aws:s3:::ReportBucket",
"Format": "Report_CSV_20180820",
"Enabled": true,
"Prefix": "reports/compliance-objects-bops",
"ReportScope": "AllTasks"
}
EOF

aws \
  s3control create-job \
  --account-id "${ACCOUNT_ID}" \
  --manifest "${MANIFEST//$'\n'}" \
  --operation "${OPERATION//$'\n'/'}" \
  --report "${REPORT//$'\n'}" \
  --priority 10 \
  --role-arn "${ROLE_ARN}" \
  --client-request-token "$(uuidgen)" \
  --region "${AWS_DEFAULT_REGION}" \
  --description "Extend compliance retention to 15 Jan 2025";

```

Verwenden des AWS SDK for Java

Example Stellen Sie den Aufbewahrungsmodus auf COMPLIANCE und das Aufbewahrungsdatum bis zum 1. Januar 2025 ein.

```

public String createComplianceRetentionJob(final AWSS3ControlClient awss3ControlClient)
    throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/compliance-objects-
manifest.csv";
    final String manifestObjectVersionId = "your-object-version-Id";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new JobManifestSpec()
            .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
            .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

```

```
final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
final String jobReportPrefix = "reports/compliance-objects-bops";

final JobReport jobReport = new JobReport()
    .withEnabled(true)
    .withReportScope(JobReportScope.AllTasks)
    .withBucket(jobReportBucketArn)
    .withPrefix(jobReportPrefix)
    .withFormat(JobReportFormat.Report_CSV_20180820);

final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
final Date janFirst = format.parse("01/01/2025");

final JobOperation jobOperation = new JobOperation()
    .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
        .withRetention(new S3Retention()
            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
            .withRetainUntilDate(janFirst)));

final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
final Boolean requiresConfirmation = true;
final int priority = 10;

final CreateJobRequest request = new CreateJobRequest()
    .withAccountId("123456789012")
    .withDescription("Set compliance retain-until to 1 Jan 2025")
    .withManifest(manifestToPublicApi)
    .withOperation(jobOperation)
    .withPriority(priority)
    .withRoleArn(roleArn)
    .withReport(jobReport)
    .withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.getJobId();
}
```

Example Erweitern des **COMPLIANCE** des **retain until date**-Modus

Im folgenden Beispiel wird das **COMPLIANCE** des **retain until date**-Modus bis zum 15. Januar 2025 erweitert.

```
public String createExtendComplianceRetentionJob(final AWSS3ControlClient
  awss3ControlClient) throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/compliance-objects-
manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
      .withObjectArn(manifestObjectArn)
      .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
      new JobManifestSpec()
        .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
        .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
      .withLocation(manifestLocation)
      .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/compliance-objects-bops";

    final JobReport jobReport = new JobReport()
      .withEnabled(true)
      .withReportScope(JobReportScope.AllTasks)
      .withBucket(jobReportBucketArn)
      .withPrefix(jobReportPrefix)
      .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan15th = format.parse("15/01/2025");

    final JobOperation jobOperation = new JobOperation()
      .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
        .withRetention(new S3Retention()
          .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
          .withRetainUntilDate(jan15th)));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
```

```
.withAccountId("123456789012")
.withDescription("Extend compliance retention to 15 Jan 2025")
.withManifest(manifestToPublicApi)
.withOperation(jobOperation)
.withPriority(priority)
.withRoleArn(roleArn)
.withReport(jobReport)
.withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.getJobId();
}
```

Verwenden von S3-Batch-Vorgänge mit dem Governance-Modus der S3-Objektsperrenaufbewahrung

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Es zeigt, wie die S3-Objektsperrenaufbewahrungs-Governance mit dem `retain until date` 30. Januar 2025 auf mehrere Objekte angewendet wird. Es erstellt einen Batch-Vorgangsauftrag, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

Verwenden Sie den AWS CLI

Example Wenden Sie die Aufbewahrungs-Governance von S3 Object Lock auf mehrere Objekte an, wobei die Aufbewahrungsfrist bis zum 30. Januar 2025 gilt

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-30T00:00:00",
      "Mode":"GOVERNANCE"
    }
  }
}
}
```

```
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/governance-objects-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucketT",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/governance-objects",
  "ReportScope": "AllTasks"
}
EOF

aws \
  s3control create-job \
  --account-id "${ACCOUNT_ID}" \
  --manifest "${MANIFEST//$'\n'}" \
  --operation "${OPERATION//$'\n'/'}" \
  --report "${REPORT//$'\n'}" \
  --priority 10 \
  --role-arn "${ROLE_ARN}" \
  --client-request-token "$(uuidgen)" \
  --region "${AWS_DEFAULT_REGION}" \
  --description "Put governance retention";
```

Example Umgehen Sie Aufbewahrungs-Governance über mehrere Objekte

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Es zeigt, wie Sie die Aufbewahrungs-Governance über mehrere Objekte hinweg umgehen und erstellt einen Batch-Vorgangsauftrag, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

```
export AWS_PROFILE='aws-user'

read -d '' bypass_governance_permissions <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:BypassGovernanceRetention"
      ],
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    }
  ]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name bypass-governance-
permissions --policy-document "${bypass_governance_permissions}"

export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "BypassGovernanceRetention": true,
    "Retention": {
    }
  }
}
}
```

```
EOF
```

```
read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3::ManifestBucket/governance-objects-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF
```

```
read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3::REPORT_BUCKET",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/bops-governance",
  "ReportScope": "AllTasks"
}
EOF
```

```
aws \
  s3control create-job \
  --account-id "${ACCOUNT_ID}" \
  --manifest "${MANIFEST//$\n'" \
  --operation "${OPERATION//$\n'/'}" \
  --report "${REPORT//$\n'" \
  --priority 10 \
  --role-arn "${ROLE_ARN}" \
  --client-request-token "$(uuidgen)" \
  --region "${AWS_DEFAULT_REGION}" \
  --description "Remove governance retention";
```


Verwenden des AWS SDK for Java

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Es zeigt, wie die Aufbewahrungs-Governance von S3 Object Lock mit einer `retain until date` Frist von bis zum 30. Januar 2025 auf mehrere Objekte angewendet werden kann. Es erstellt einen Batch-Vorgangsauftrag, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

Example Wenden Sie die Aufbewahrungs-Governance von S3 Object Lock auf mehrere Objekte an, wobei die Aufbewahrungsfrist bis zum 30. Januar 2025 gilt

```
public String createGovernanceRetentionJob(final AWSS3ControlClient awss3ControlClient)
    throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/governance-objects-
manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new JobManifestSpec()
            .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
            .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/governance-objects";

    final JobReport jobReport = new JobReport()
        .withEnabled(true)
        .withReportScope(JobReportScope.AllTasks)
        .withBucket(jobReportBucketArn)
        .withPrefix(jobReportPrefix)
        .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan30th = format.parse("30/01/2025");
```

```

final JobOperation jobOperation = new JobOperation()
    .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
        .withRetention(new S3Retention()
            .withMode(S3ObjectLockRetentionMode.GOVERNANCE)
            .withRetainUntilDate(jan30th)));

final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
final Boolean requiresConfirmation = true;
final int priority = 10;

final CreateJobRequest request = new CreateJobRequest()
    .withAccountId("123456789012")
    .withDescription("Put governance retention")
    .withManifest(manifestToPublicApi)
    .withOperation(jobOperation)
    .withPriority(priority)
    .withRoleArn(roleArn)
    .withReport(jobReport)
    .withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.getJobId();
}

```

Example Umgehen Sie Aufbewahrungs-Governance über mehrere Objekte

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Es zeigt, wie Sie die Aufbewahrungs-Governance über mehrere Objekte hinweg umgehen und erstellt einen Batch-Vorgangsauftrag, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

```

public void allowBypassGovernance() {
    final String roleName = "bops-object-lock";

    final String bypassGovernancePermissions = "{" +
        "  \"Version\": \"2012-10-17\", " +
        "  \"Statement\": [" +
        "    {" +
        "      \"Effect\": \"Allow\", " +
        "      \"Action\": [" +

```

```

        "                \"s3:BypassGovernanceRetention\"" +
        "            ]," +
        "            \"Resource\": [" +
        "                \"arn:aws:s3:::ManifestBucket/*\"" +
        "            ]" +
        "        }" +
        "    ]" +
        "};

final AmazonIdentityManagement iam =
    AmazonIdentityManagementClientBuilder.defaultClient();

final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
    .withPolicyDocument(bypassGovernancePermissions)
    .withPolicyName("bypass-governance-permissions")
    .withRoleName(roleName);

final PutRolePolicyResult putRolePolicyResult =
iam.putRolePolicy(putRolePolicyRequest);
}
public String createRemoveGovernanceRetentionJob(final AWSS3ControlClient
awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/governance-objects-
manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
        .withObjectArn(manifestObjectArn)
        .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
        new JobManifestSpec()
            .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
            .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
        .withLocation(manifestLocation)
        .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/bops-governance";

    final JobReport jobReport = new JobReport()
        .withEnabled(true)

```

```
.withReportScope(JobReportScope.AllTasks)
.withBucket(jobReportBucketArn)
.withPrefix(jobReportPrefix)
.withFormat(JobReportFormat.Report_CSV_20180820);

final JobOperation jobOperation = new JobOperation()
    .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
        .withRetention(new S3Retention()));

final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
final Boolean requiresConfirmation = true;
final int priority = 10;

final CreateJobRequest request = new CreateJobRequest()
    .withAccountId("123456789012")
    .withDescription("Remove governance retention")
    .withManifest(manifestToPublicApi)
    .withOperation(jobOperation)
    .withPriority(priority)
    .withRoleArn(roleArn)
    .withReport(jobReport)
    .withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);

return result.getJobId();
}
```

Verwendung rechtlicher Aufbewahrungsfristen für die S3-Objektsperre mithilfe von S3-Batch-Vorgänge deaktivieren

Das folgende Beispiel baut auf den vorherigen Beispielen zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Es zeigt, wie Sie die rechtliche Aufbewahrungsfrist für die Objektsperre für Objekte mithilfe von Batch-Vorgänge deaktivieren können.

Im Beispiel wird zuerst die Rolle aktualisiert, um `s3:PutObjectLegalHold`-Berechtigungen zu erteilen, ein Batch-Vorgangsauftrag erstellt, der die rechtliche Aufbewahrungsfrist von den im Manifest identifizierten Objekten deaktiviert (entfernt) und anschließend Berichte darüber erstellt.

Mit dem AWS CLI

Example Aktualisiert die Rolle, um **s3:PutObjectLegalHold** Berechtigungen zu erteilen

```
export AWS_PROFILE='aws-user'

read -d '' legal_hold_permissions <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectLegalHold"
      ],
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    }
  ]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name legal-hold-
permissions --policy-document "${legal_hold_permissions}"
```

Example Deaktivieren der gesetzlichen Aufbewahrungsfrist

Im folgenden Beispiel wird die rechtliche Aufbewahrungsfrist deaktiviert.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectLegalHold": {
    "LegalHold": {
      "Status":"OFF"
    }
  }
}
}
```

```
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/legalhold-object-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/legalhold-objects-bops",
  "ReportScope": "AllTasks"
}
EOF

aws \
  s3control create-job \
  --account-id "${ACCOUNT_ID}" \
  --manifest "${MANIFEST//$'\n'}" \
  --operation "${OPERATION//$'\n'/'}" \
  --report "${REPORT//$'\n'}" \
  --priority 10 \
  --role-arn "${ROLE_ARN}" \
  --client-request-token "$(uuidgen)" \
  --region "${AWS_DEFAULT_REGION}" \
  --description "Turn off legal hold";
```

Verwenden des AWS SDK for Java

Example Aktualisiert die Rolle, um **s3:PutObjectLegalHold** Berechtigungen zu erteilen

```
public void allowPutObjectLegalHold() {
    final String roleName = "bops-object-lock";

    final String legalHoldPermissions = "{" +
        "  \"Version\": \"2012-10-17\", " +
        "  \"Statement\": [" +
        "    {" +
        "      \"Effect\": \"Allow\", " +
        "      \"Action\": [" +
        "        \"s3:PutObjectLegalHold\" " +
        "      ], " +
        "      \"Resource\": [" +
        "        \"arn:aws:s3:::ManifestBucket/*\" " +
        "      ] " +
        "    } " +
        "  ] " +
        "};";

    final AmazonIdentityManagement iam =
        AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
        .withPolicyDocument(legalHoldPermissions)
        .withPolicyName("legal-hold-permissions")
        .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
        iam.putRolePolicy(putRolePolicyRequest);
}
```

Example Deaktivieren der gesetzlichen Aufbewahrungsfrist

Verwenden Sie das folgende Beispiel, wenn Sie die rechtliche Aufbewahrungsfrist deaktivieren möchten.

```
public String createLegalHoldOffJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/legalhold-object-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";
```

```
final JobManifestLocation manifestLocation = new JobManifestLocation()
    .withObjectArn(manifestObjectArn)
    .withETag(manifestObjectVersionId);

final JobManifestSpec manifestSpec =
    new JobManifestSpec()
        .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
        .withFields("Bucket", "Key");

final JobManifest manifestToPublicApi = new JobManifest()
    .withLocation(manifestLocation)
    .withSpec(manifestSpec);

final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
final String jobReportPrefix = "reports/legalhold-objects-bops";

final JobReport jobReport = new JobReport()
    .withEnabled(true)
    .withReportScope(JobReportScope.AllTasks)
    .withBucket(jobReportBucketArn)
    .withPrefix(jobReportPrefix)
    .withFormat(JobReportFormat.Report_CSV_20180820);

final JobOperation jobOperation = new JobOperation()
    .withS3PutObjectLegalHold(new S3SetObjectLegalHoldOperation()
        .withLegalHold(new S3ObjectLockLegalHold()
            .withStatus(S3ObjectLockLegalHoldStatus.OFF)));

final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
final Boolean requiresConfirmation = true;
final int priority = 10;

final CreateJobRequest request = new CreateJobRequest()
    .withAccountId("123456789012")
    .withDescription("Turn off legal hold")
    .withManifest(manifestToPublicApi)
    .withOperation(jobOperation)
    .withPriority(priority)
    .withRoleArn(roleArn)
    .withReport(jobReport)
    .withConfirmationRequired(requiresConfirmation);

final CreateJobResult result = awss3ControlClient.createJob(request);
```



```
    return result.getJobId();  
}
```

Tutorial zu S3-Batchvorgängen

Im folgenden Tutorial werden vollständige End-to-End-Verfahren für einige Batchvorgänge vorgestellt.

- [Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations, AWS Lambda, und AWS Elemental MediaConvert](#)

Überwachen von Amazon S3

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon S3 und Ihrer AWS Lösungen. Wir empfehlen, Überwachungsdaten aus allen Teilen Ihres Systems zu sammeln AWS Lösung, damit Sie einen Mehrpunktfehler einfacher debuggen können, falls einer auftritt. Bevor Sie mit der Überwachung von Amazon S3 beginnen, sollten Sie einen Überwachungsplan mit Antworten auf die folgenden Fragen erstellen:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungstools werden verwendet?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Weitere Informationen zur Protokollierung und Überwachung in Amazon S3 finden Sie in den folgenden Themen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Überwachungstools](#)
- [Protokollierungsoptionen für Amazon S3](#)
- [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#)
- [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#)
- [Metriken mit Amazon überwachen CloudWatch](#)
- [Amazon-S3-Ereignis-Benachrichtigungen](#)

Überwachungstools

AWS bietet verschiedene Tools, mit denen Sie Amazon S3 überwachen können. Sie können einige dieser Tools so konfigurieren, dass diese die Überwachung für Sie übernehmen, während bei anderen Tools ein manuelles Eingreifen nötig ist. Wir empfehlen, dass Sie die Überwachungsaufgaben möglichst automatisieren.

Automatisierte Überwachungstools

Sie können die folgenden automatisierten Tools zur Überwachung von Amazon S3 verwenden und auftretende Probleme melden:

- Amazon CloudWatch Alarms — Überwachen Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum und führen Sie eine oder mehrere Aktionen aus, die auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über mehrere Zeiträume basieren. Die Aktion ist eine Benachrichtigung, die an ein Amazon Simple Notification Service (AmazonSNS) -Thema oder eine Amazon EC2 Auto Scaling Scaling-Richtlinie gesendet wird. CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Status befinden. Der Status muss sich geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).
- AWS CloudTrail Protokollüberwachung — Teilen Sie Protokolldateien zwischen Konten, überwachen CloudTrail Sie Protokolldateien in Echtzeit, indem Sie sie an CloudWatch Logs senden, schreiben Sie Protokollverarbeitungsanwendungen in Java und überprüfen Sie, ob sich Ihre Protokolldateien nach der Lieferung von CloudTrail nicht geändert haben. Weitere Informationen finden Sie unter [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#).

Manuelle Überwachungstools

Ein weiterer wichtiger Teil der Überwachung von Amazon S3 ist die manuelle Überwachung der Elemente, die von den CloudWatch Alarmen nicht abgedeckt werden. Der Amazon S3 CloudWatch, Trusted Advisor, und andere AWS Management Console Dashboards bieten einen at-a-glance Überblick über den Status Ihrer AWS Umgebung. Möglicherweise wollen Sie die Server-Zugriffsprotokollierung aktivieren, die Zugriffsanforderungen für den Bucket verfolgt. Jeder Zugriffsprotokolldatensatz enthält Details über eine Zugriffsanforderung, z. B. Auftraggeber, Bucket-Name, Anforderungszeit, Anforderungsaktion, Antwortstatus und Fehlercode, falls vorhanden. Weitere Informationen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

- Auf dem Amazon-S3-Dashboard wird Folgendes angezeigt:
 - Ihre Buckets und die Eigenschaften und Objekte, die sie enthalten.
- Die CloudWatch Startseite zeigt Folgendes:
 - Aktuelle Alarmer und Status
 - Diagramme mit Alarmen und Ressourcen
 - Servicestatus

Darüber hinaus können CloudWatch Sie Folgendes verwenden:

- Erstellen von [benutzerdefinierten Dashboards](#) zur Überwachung des gewünschten Services.
- Aufzeichnen von Metrikdaten, um Probleme zu beheben und Trends zu erkennen.
- Suchen und durchsuchen Sie all Ihre AWS Ressourcen-Metriken.
- Erstellen und Bearbeiten von Alarmen, um über Probleme benachrichtigt zu werden.
- AWS Trusted Advisor kann Ihnen helfen, Ihre zu überwachen AWS Ressourcen zur Verbesserung von Leistung, Zuverlässigkeit, Sicherheit und Wirtschaftlichkeit. Vier Trusted Advisor Checks stehen allen Benutzern zur Verfügung; mehr als 50 Checks stehen Benutzern mit einem Business- oder Enterprise-Supportplan zur Verfügung. Weitere Informationen finden Sie unter [AWS Trusted Advisor](#).

Trusted Advisor hat diese Prüfungen, die sich auf Amazon S3 beziehen:

- Überprüfungen der Protokollierungskonfiguration von Amazon-S3-Buckets.
- Sicherheitsprüfungen für Amazon-S3-Buckets mit offenen Zugriffsberechtigungen.
- Fehlertoleranzprüfungen für Amazon-S3-Buckets, für die kein Versioning aktiviert oder deren Versioning ausgesetzt ist.

Protokollierungsoptionen für Amazon S3

Sie können die Aktionen aufzeichnen, die von Benutzern, Rollen oder AWS-Services auf Amazon S3 S3-Ressourcen und zur Aufbewahrung von Protokollaufzeichnungen zu Prüfungs- und Compliance-Zwecken. Zu diesem Zweck können Sie die Protokollierung des Serverzugriffs verwenden, AWS CloudTrail Protokollierung oder eine Kombination aus beidem. Wir empfehlen Ihnen, CloudTrail für die Protokollierung von Aktionen auf Bucket- und Objektebene für Ihre Amazon S3 S3-Ressourcen zu verwenden. Weitere Informationen zu jeder Option finden Sie in folgenden Abschnitten:

- [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#)

- [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#)

In der folgenden Tabelle sind die wichtigsten Eigenschaften von CloudTrail Protokollen und Amazon S3-Serverzugriffsprotokollen aufgeführt. Sehen Sie sich die Tabelle und die Hinweise an, um sicherzustellen, dass dies Ihren Sicherheitsanforderungen CloudTrail entspricht.

Protokolleigenschaften	AWS CloudTrail	Amazon-S3-Serverprotokolle
Kann an andere Systeme weitergeleitet werden (Amazon CloudWatch Logs, Amazon CloudWatch Events)	Ja	Nein
Protokolle an mehreren Zielen bereitstellen (Beispiel: dieselben Protokolle an zwei verschiedene Buckets senden)	Ja	Nein
Protokolle für eine Teilmenge von Objekten aktivieren (Präfix)	Ja	Nein
Kontoübergreifende Protokollbereitstellung (Ziel- und Quell-Bucket in Besitz von verschiedenen Konten)	Ja	Nein
Integritätsprüfung der Protokolldatei mit digitaler Signatur oder Hashing	Ja	Nein
Standardeinstellung oder Auswahl der Verschlüsselung für Protokolldateien	Ja	Nein
Objektoperationen (mithilfe von Amazon S3 APIs)	Ja	Ja

Protokolleigenschaften	AWS CloudTrail	Amazon-S3-Serverprotokolle
Bucket-Operationen (mithilfe von Amazon S3 APIs)	Ja	Ja
Durchsuchbare UI für Protokolle	Ja	Nein
Felder für Objektsperreparameter, Amazon S3 Select-Eigenschaften für Protokolldatensätze	Ja	Nein
Felder für Object Size, Total Time, Turn-Around Time und HTTP Referer für Protokolldatensätze	Nein	Ja
Lebenszyklusübertragungen, Ablaufaktionen, Wiederherstellungen	Nein	Ja
Protokollieren von Schlüssel in einer Batch-Delete-Operation	Nein	Ja
Authentifizierungsfehler ¹	Nein	Ja
Konten, an die Protokolle geliefert werden	Bucket-Eigentümer ² und Auftraggeber	Nur Bucket-Eigentümer
Performance and Cost	AWS CloudTrail	Amazon S3 Server Logs
Preis	Verwaltungsereignisse (erste Bereitstellung) sind kostenlos. Für Datenereignisse fällt zusätzlich zur Speicherung der Protokolle eine Gebühr an	Keine Zusatzkosten neben der Speicherung der Protokolle

Protokolleigenschaften	AWS CloudTrail	Amazon-S3-Serverprotokolle
Geschwindigkeit der Protokollbereitstellung	Datenereignisse alle 5 Minuten; Verwaltungsereignisse alle 15 Minuten	Innerhalb weniger Stunden
Protokollformat	JSON	Protokolldatei mit durch Leerzeichen getrennten, durch neue Zeilen getrennten Datensätzen

Hinweise

1. CloudTrail liefert keine Protokolle für Anfragen, bei denen die Authentifizierung fehlschlägt (bei denen die angegebenen Anmeldeinformationen nicht gültig sind). Es enthält jedoch Protokolle für Anforderungen, deren Authentifizierung fehlschlägt (AccessDenied), und Anforderungen, die von anonymen Benutzern gestellt werden.
2. Der S3-Bucket-Besitzer erhält CloudTrail Protokolle, wenn das Konto keinen vollen Zugriff auf das Objekt in der Anfrage hat. Weitere Informationen finden Sie unter [Amazon S3 S3-Aktionen auf Objektebene in kontoübergreifenden Szenarien](#).
3. S3 unterstützt nicht die Übermittlung von CloudTrail Protokollen oder Serverzugriffsprotokollen an den Anforderer oder den Bucket-Besitzer für VPC Endpunktanfragen, wenn die VPC Endpunktrichtlinie sie ablehnt, oder für Anfragen, die fehlschlagen, bevor die VPC Richtlinie ausgewertet wurde.

Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail

Amazon S3 ist integriert mit [AWS CloudTrail](#), ein Service, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS-Service. CloudTrail erfasst alle API Aufrufe für Amazon S3 als Ereignisse. Zu den erfassten Anrufen gehören Aufrufe von der Amazon S3 S3-Konsole und Code-Aufrufe an die Amazon S3 API S3-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon S3 gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Ob die Anfrage im Namen eines IAM Identity Center-Benutzers gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen gestellt wurde AWS-Service.

CloudTrail ist aktiv in deinem AWS-Konto wenn Sie das Konto erstellen und Sie automatisch Zugriff auf den CloudTrail Event-Verlauf haben. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem AWS-Region. Weitere Informationen finden Sie unter [Arbeiten mit dem CloudTrail Ereignisverlauf](#) in der AWS CloudTrail Benutzerleitfaden. Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto Erstellen Sie einen Trail- oder [CloudTrailLake-Event-Datenspeicher](#) der letzten 90 Tage.

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle Pfade wurden mit dem erstellt AWS Management Console sind regionsübergreifend. Sie können einen Pfad mit einer oder mehreren Regionen erstellen, indem Sie AWS CLI. Es wird empfohlen, einen Wanderweg mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten erfassen AWS-Regionen in deinem Konto. Wenn du einen Trail mit nur einer Region erstellst, kannst du dir nur die Ereignisse ansehen, die in den Trails protokolliert wurden AWS-Region. Weitere Informationen zu Pfaden findest du unter [Einen Trail für dich erstellen AWS-Kontound](#) [Einen Trail für eine Organisation](#) erstellen in der AWS CloudTrail Benutzerleitfaden.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preisgestaltung](#). Informationen zu Amazon-S3-Preisen finden Sie unter [Amazon S3-Preise](#).

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL basierte Abfragen zu Ihren Ereignissen ausführen. CloudTrail Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON Format in das [ORCApache-Format](#). ORCist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst,

bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von [erweiterten Ereignisselektoren](#) auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter [Arbeiten mit AWS CloudTrail See](#) im AWS CloudTrail Benutzerleitfaden.

CloudTrail Für Datenspeicher und Abfragen von Ereignissen in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die [Preisoption](#) aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preisgestaltung](#).

Sie können Ihre Protokolldateien beliebig lange im Bucket speichern. Sie können aber auch Amazon S3-Lebenszyklusregeln aufstellen, anhand derer die Protokolldateien automatisch archiviert oder gelöscht werden. Standardmäßig werden Ihre Protokolldateien mithilfe der serverseitigen Amazon S3 S3-Verschlüsselung (SSE) verschlüsselt.


Verwenden von CloudTrail Protokollen mit Amazon S3 S3-Serverzugriffsprotokollen und CloudWatch Protokollen

AWS CloudTrail Protokolle enthalten eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon S3, während Amazon S3 S3-Serverzugriffsprotokolle detaillierte Aufzeichnungen über die Anfragen enthalten, die an einen S3-Bucket gestellt werden. Weitere Informationen zur Funktionsweise der unterschiedlichen Protokolle und ihren Eigenschaften, ihrer Leistung und ihrer Kosten finden Sie unter [the section called "Protokollierungsoptionen"](#).

Sie können Folgendes verwenden ... AWS CloudTrail protokolliert zusammen mit Serverzugriffsprotokollen für Amazon S3. CloudTrail Protokolle bieten Ihnen eine detaillierte API Nachverfolgung von Amazon S3 S3-Vorgängen auf Bucket- und Objektebene. Serverzugriffsprotokolle für Amazon S3 bieten Ihnen Einblick in Vorgänge auf Objektebene mit Ihren Daten in Amazon S3. Weitere Informationen zu Server-Zugriffsprotokollen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

Sie können CloudTrail Protokolle auch zusammen mit Amazon CloudWatch für Amazon S3 verwenden. CloudTrail Durch die Integration mit CloudWatch Logs werden API Aktivitäten auf S3-Bucket-Ebene, die von erfasst wurden CloudTrail , in einem CloudWatch Protokollstream in der von Ihnen CloudWatch angegebenen Protokollgruppe bereitgestellt. Sie können CloudWatch Alarmer zur


Überwachung bestimmter API Aktivitäten erstellen und E-Mail-Benachrichtigungen erhalten, wenn die bestimmte API Aktivität auftritt. Weitere Informationen zu CloudWatch Alarmen zur Überwachung bestimmter API Aktivitäten finden Sie im [AWS CloudTrail Benutzerleitfaden](#). Weitere Informationen zur Verwendung CloudWatch mit Amazon S3 finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

 Note

S3 unterstützt nicht die Übermittlung von CloudTrail Protokollen an den Anforderer oder den Bucket-Besitzer für VPC Endpunktanfragen, wenn die VPC Endpunktrichtlinie diese ablehnt.

CloudTrail Nachverfolgung mit Amazon S3 SOAP API S3-Anrufen

CloudTrail verfolgt Amazon S3 SOAP API S3-Anrufe. Die Amazon S3 SOAP S3-Unterstützung HTTP ist veraltet, aber sie ist weiterhin verfügbar. HTTPS Weitere Informationen zur Amazon S3 SOAP S3-Unterstützung finden Sie unter [Anhang A: Verwenden der SOAP-API](#).

 Important

Neuere Amazon S3 S3-Funktionen werden nicht unterstützt SOAP. Wir empfehlen Ihnen, entweder den REST API oder den zu verwenden AWS SDKs.

Amazon S3 SOAP S3-Aktionen, die durch CloudTrail Protokollierung verfolgt werden

SOAPAPIName	APIName des Ereignisses, der im CloudTrail Protokoll verwendet wird
ListAllMyBuckets	ListBuckets
CreateBucket	CreateBucket
DeleteBucket	DeleteBucket
GetBucketAccessControlPolicy	GetBucketAc1
SetBucketAccessControlPolicy	PutBucketAc1
GetBucketLoggingStatus	GetBucketLogging

SOAPAPIName	APIName des Ereignisses, der im CloudTrail Protokoll verwendet wird
SetBucketLoggingStatus	PutBucketLogging

Weitere Informationen zu Amazon S3 CloudTrail und Amazon S3 finden Sie in den folgenden Themen:

Themen

- [Amazon S3 CloudTrail S3-Ereignisse](#)
- [CloudTrail Protokolldateieinträge für Amazon S3 und S3 auf Outposts](#)
- [Aktivieren der CloudTrail Ereignisprotokollierung für S3-Buckets und -Objekte](#)
- [Identifizieren von Amazon S3 S3-Anfragen mit CloudTrail](#)

Amazon S3 CloudTrail S3-Ereignisse

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Dieser Abschnitt enthält Informationen zu den Ereignissen, die S3 protokolliert CloudTrail.

Amazon S3 S3-Datenergebnisse in CloudTrail

[Datenergebnisse](#) liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. Lesen oder Schreiben in ein Amazon-S3-Objekt). Sie

werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Protokolliert standardmäßig CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preisgestaltung](#).

Sie können Datenereignisse für die Amazon S3 S3-Ressourcentypen mithilfe der CloudTrail Konsole protokollieren. AWS CLI, oder CloudTrail API Operationen. Weitere Informationen zum Protokollieren von Datenereignissen finden Sie unter [Protokollieren von Datenereignissen mit dem AWS Management Console](#) und [Protokollieren von Datenereignissen mit dem AWS Command Line Interface](#) in der AWS CloudTrail Benutzerleitfaden.

In der folgenden Tabelle sind die Amazon S3 S3-Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, den Sie aus der Liste Datenereignistyp auf der CloudTrail Konsole auswählen können. In der Wertspalte resources.type wird der resources.type Wert angezeigt, den Sie angeben würden, wenn Sie erweiterte Event-Selektoren mit dem AWS CLI oder CloudTrail APIs In der CloudTrail Spalte APIsProtokollierte Daten werden die API Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3	AWS::S3::Object	<ul style="list-style-type: none"> • AbortMultipartUpload • CompleteMultipartUpload • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • DeleteObjects • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
		<ul style="list-style-type: none">• GetObjectTagging• GetObjectTorrent• HeadObject• ListMultipartUploads• ListObjectVersions• ListObjects• ListParts• PutObject• PutObjectAcl• PutObjectLegalHold• PutObjectRetention• PutObjectTagging• RestoreObject• SelectObjectContent• UploadPart• UploadPartCopy

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3 Express Eine Zone	AWS::S3Express::Object	<ul style="list-style-type: none">• AbortMultipartUpload• CompleteMultipartUpload• CreateSession• CopyObject• CreateMultipartUpload• DeleteObject• DeleteObjects• GetObject• GetObjectAttributes• HeadBucket• HeadObject• ListObjectsV2• ListParts• PutObject• UploadPart• UploadPartCopy

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3-Zugangspunkt	AWS::S3::Access Point	<ul style="list-style-type: none"> • AbortMultipartUpload • CompleteMultipartUpload • CopyObject (nur Kopien in derselben Region) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetBucketAcl • GetBucketCors • GetBucketLocation • GetBucketNotificationConfiguration • GetBucketPolicy • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadBucket • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListObjectVersions • ListParts • Presign • PutObject

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
		<ul style="list-style-type: none">• PutObjectLegalHold• PutObjectRetention• PutObjectAcl• PutObjectTagging• RestoreObject• UploadPart• UploadPartCopy (nur Kopien in derselben Region)

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3 Objekt Lambda	AWS::S3objectLambda::AccessPoint	<ul style="list-style-type: none"> • AbortMultipartUpload • CompleteMultipartUpload • CopyObject (nur Kopien in derselben Region) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectAcl • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectVersions • ListParts • PutObject • PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • WriteGetObjectResponse

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3-Outposts	AWS::S3Outposts::Object	<ul style="list-style-type: none"> • AbortMultipartUpload • CompleteMultipartUpload • CopyObject (nur Kopien in derselben Region) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListParts • PutObject • PutObjectTagging • UploadPart • UploadPartCopy

Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den `resources.ARN` Feldern `eventNameReadOnly`, und filtern, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Feldern finden Sie unter [AdvancedFieldSelector](#) in der AWS CloudTrail APIReferenz.

Amazon S3 S3-Managementereignisse in CloudTrail

Amazon S3 protokolliert alle Operationen auf der Kontrollebene als Verwaltungsereignisse. Weitere Informationen zu API S3-Vorgängen finden Sie in der [Amazon S3 API S3-Referenz](#).

So werden Anfragen an Amazon S3 CloudTrail erfasst

Standardmäßig CloudTrail protokolliert API S3-Aufrufe auf Bucket-Ebene, die in den letzten 90 Tagen getätigt wurden, protokolliert jedoch keine Anfragen an Objekte. Aufrufe auf Bucket-Ebene sind Ereignisse wie `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy` usw. Sie können Ereignisse auf Bucket-Ebene auf der Konsole sehen. CloudTrail Sie können dort jedoch keine Datenereignisse (Amazon S3 S3-Aufrufe auf Objektebene) anzeigen — Sie müssen sie analysieren oder die Protokolle danach abfragen. CloudTrail

Amazon S3 S3-Aktionen auf Kontoebene, die durch Protokollierung verfolgt werden CloudTrail

CloudTrail protokolliert Aktionen auf Kontoebene. Amazon S3 S3-Datensätze werden zusammen mit anderen geschriebenen AWS-Service Aufzeichnungen in einer Protokolldatei. CloudTrail bestimmt anhand eines Zeitraums und der Dateigröße, wann eine neue Datei erstellt und in diese geschrieben werden soll.

In den Tabellen in diesem Abschnitt sind die Amazon S3 S3-Aktionen auf Kontoebene aufgeführt, die für die Protokollierung von unterstützt werden. CloudTrail

APIAktionen auf Amazon S3 S3-Kontoebene, die durch CloudTrail Protokollierung verfolgt werden, werden als die folgenden Ereignisnamen angezeigt. Die Namen der CloudTrail Ereignisse unterscheiden sich vom API Aktionsnamen. Zum Beispiel `DeletePublicAccessBlock` ist `DeleteAccountPublicAccessBlock`.

- [DeleteAccountPublicAccessBlock](#)
- [GetAccountPublicAccessBlock](#)
- [PutAccountPublicAccessBlock](#)

Amazon S3 S3-Aktionen auf Bucket-Ebene, die durch Protokollierung verfolgt werden CloudTrail

Standardmäßig werden Aktionen auf Bucket-Ebene für allgemeine Buckets CloudTrail protokolliert. Amazon S3 S3-Datensätze werden zusammen mit anderen geschriebenen AWS Serviceaufzeichnungen in einer Protokolldatei. CloudTrail bestimmt anhand eines Zeitraums und der Dateigröße, wann eine neue Datei erstellt und in diese geschrieben werden soll.

In diesem Abschnitt sind die Amazon S3 S3-Aktionen auf Bucket-Ebene aufgeführt, die für die Protokollierung von unterstützt werden. CloudTrail

APIAktionen auf Amazon S3 S3-Bucket-Ebene, die durch CloudTrail Protokollierung verfolgt werden, werden als die folgenden Ereignisnamen angezeigt. In einigen Fällen unterscheidet sich der Name des CloudTrail Ereignisses vom API Aktionsnamen. Zum Beispiel, PutBucketLifecycleConfiguration ist PutBucketLifecycle.

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketAnalyticsConfiguration](#)
- [DeleteBucketCors](#)
- [DeleteBucketEncryption](#)
- [DeleteBucketIntelligentTieringConfiguration](#)
- [DeleteBucketInventoryConfiguration](#)
- [DeleteBucketLifecycle](#)
- [DeleteBucketMetricsConfiguration](#)
- [DeleteBucketOwnershipControls](#)
- [DeleteBucketPolicy](#)
- [DeleteBucketPublicAccessBlock](#)
- [DeleteBucketReplication](#)
- [DeleteBucketTagging](#)
- [GetAccelerateConfiguration](#)
- [GetBucketAcl](#)
- [GetBucketAnalyticsConfiguration](#)
- [GetBucketCors](#)
- [GetBucketEncryption](#)
- [GetBucketIntelligentTieringConfiguration](#)
- [GetBucketInventoryConfiguration](#)
- [GetBucketLifecycle](#)
- [GetBucketLocation](#)
- [GetBucketLogging](#)

- [GetBucketMetricsConfiguration](#)
- [GetBucketNotification](#)
- [GetBucketObjectLockConfiguration](#)
- [GetBucketOwnershipControls](#)
- [GetBucketPolicy](#)
- [GetBucketPolicyStatus](#)
- [GetBucketPublicAccessBlock](#)
- [GetBucketReplication](#)
- [GetBucketRequestPayment](#)
- [GetBucketTagging](#)
- [GetBucketVersioning](#)
- [GetBucketWebsite](#)
- [HeadBucket](#)
- [ListBuckets](#)
- [PutAccelerateConfiguration](#)
- [PutBucketAcl](#)
- [PutBucketAnalyticsConfiguration](#)
- [PutBucketCors](#)
- [PutBucketEncryption](#)
- [PutBucketIntelligentTieringConfiguration](#)
- [PutBucketInventoryConfiguration](#)
- [PutBucketLifecycle](#)
- [PutBucketLogging](#)
- [PutBucketMetricsConfiguration](#)
- [PutBucketNotification](#)
- [PutBucketObjectLockConfiguration](#)
- [PutBucketOwnershipControls](#)
- [PutBucketPolicy](#)
- [PutBucketPublicAccessBlock](#)

- [PutBucketReplication](#)
- [PutBucketRequestPayment](#)
- [PutBucketTagging](#)
- [PutBucketVersioning](#)
- [PutBucketWebsite](#)

Zusätzlich zu diesen API Operationen können Sie auch die [OPTIONS Aktion auf](#) Objektebene. Diese Aktion wird in der CloudTrail Protokollierung wie eine Aktion auf Bucket-Ebene behandelt, da die Aktion die CORS Konfiguration eines Buckets überprüft.

Amazon S3 Express One Zone-Aktionen auf Bucket-Ebene (regionaler API Endpunkt), die durch Protokollierung verfolgt werden CloudTrail

Standardmäßig werden Aktionen auf Bucket-Ebene für Verzeichnis-Buckets als Verwaltungsereignisse CloudTrail protokolliert. Die eventsource für CloudTrail Verwaltungsereignisse für S3 Express One Zone ist. `s3express.amazonaws.com`

Die folgenden regionalen API Endpunktvorgänge werden protokolliert CloudTrail.


- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketPolicy](#)
- [GetBucketPolicy](#)
- [PutBucketPolicy](#)
- [ListDirectoryBuckets](#)
- [PutBucketEncryption](#)
- [GetBucketEncryption](#)
- [DeleteBucketEncryption](#)

Weitere Informationen finden Sie unter [Protokollierung mit AWS CloudTrail für S3 Express One Zone](#)

Amazon S3 S3-Aktionen auf Objektebene in kontoübergreifenden Szenarien

Im Folgenden finden Sie spezielle Anwendungsfälle, die API Aufrufe auf Objektebene in kontoübergreifenden Szenarien und die Art und Weise, wie Protokolle gemeldet werden, betreffen.

CloudTrail CloudTrail übermittelt Protokolle an den Anforderer (das Konto, das den API Anruf getätigt hat), außer in Fällen, in denen der Zugriff verweigert wurde, in denen Protokolleinträge geschwärzt oder weggelassen werden. Bei der Einrichtung von kontoübergreifendem Zugriff sehen Sie sich die Beispiele in diesem Abschnitt an.


 Note

In den Beispielen wird davon ausgegangen, dass die CloudTrail Protokolle entsprechend konfiguriert sind.

Beispiel 1: CloudTrail Liefert Logs an den Bucket-Besitzer

CloudTrail übermittelt Protokolle an den Bucket-Besitzer, auch wenn der Bucket-Besitzer keine Berechtigungen für dieselbe API Objektoperation besitzt. Sehen Sie sich das folgende kontenübergreifende Szenario vor:

- Konto A ist Eigentümer des Buckets.
- Konto B (Anforderer) versucht, auf ein Objekt in diesem Bucket zuzugreifen.
- Konto C besitzt das Objekt. Konto C kann dasselbe Konto wie Konto A sein oder auch nicht.

 Note

CloudTrail übermittelt dem Anforderer (Konto B) immer API Logs auf Objektebene. Darüber hinaus werden dieselben Protokolle CloudTrail auch dann an den Bucket-Besitzer (Konto A) übermittelt, wenn der Bucket-Besitzer das Objekt (Konto C) nicht besitzt oder keine Berechtigungen für dieselben API Operationen an diesem Objekt besitzt.

Beispiel 2: E-Mail-Adressen, die bei der Objekteinstellung verwendet werden, werden CloudTrail nicht häufig verwendet ACLs

Sehen Sie sich das folgende kontenübergreifende Szenario vor:

- Konto A ist Eigentümer des Buckets.
- Konto B (der Antragsteller) sendet mithilfe einer E-Mail-Adresse eine Anfrage zur Einrichtung ACL einer Objektzuweisung. Weitere Informationen zu finden Sie ACLs unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

Der Anforderer erhält die Protokolle zusammen mit der E-Mail-Information. Der Bucket-Besitzer erhält jedoch — sofern er berechtigt ist, Logs zu empfangen, wie in Beispiel 1 — das CloudTrail Protokoll, das das Ereignis meldet. Der Bucket-Besitzer erhält jedoch nicht die ACL Konfigurationsinformationen, insbesondere die E-Mail-Adresse des Empfängers und den Zuschuss. Die einzige Information, die das Protokoll dem Bucket-Besitzer mitteilt, ist, dass ein ACL API Anruf von Konto B getätigt wurde.

CloudTrail Protokolldateieinträge für Amazon S3 und S3 auf Outposts

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist verfügbar in AWS CloudTrail Logs, S3 Inventory, S3 Storage Lens, die Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in der AWS Command Line Interface and AWS SDKs. Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Note

Beispiele für CloudTrail Protokolldateien für Amazon S3 Express One Zone finden Sie unter [Beispiele für CloudTrail Protokolldateien für S3 Express One Zone](#).

Weitere Informationen finden Sie in den folgenden Beispielen.

Themen

- [Beispiel: CloudTrail Protokolldateieintrag für Amazon S3](#)
- [Beispiel: Amazon S3 in Outposts-Logdateieinträgen](#)

Beispiel: CloudTrail Protokolldateieintrag für Amazon S3

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die [GET Dienst](#), [PutBucketAcl](#), und [GetBucketVersioning](#)Aktionen.

```
{
  "Records": [
    {
      "eventVersion": "1.03",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/myUserName",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "myUserName"
      },
      "eventTime": "2019-02-01T03:18:19Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "ListBuckets",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "[]",
      "requestParameters": {
        "host": [
          "s3.us-west-2.amazonaws.com"
        ]
      },
      "responseElements": null,
      "additionalEventData": {
        "SignatureVersion": "SigV2",
        "AuthenticationMethod": "QueryString",
        "aclRequired": "Yes"
      },
    },
    {
      "requestID": "47B8E8D397DCE7A6",
      "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
      "eventType": "AwsApiCall",
      "recipientAccountId": "444455556666",
      "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "s3.amazonaws.com"
      }
    }
  ]
}
```

```

},
{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/myUserName",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "myUserName"
  },
  "eventTime": "2019-02-01T03:22:33Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "PutBucketAcl",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "",
  "userAgent": "[]",
  "requestParameters": {
    "bucketName": "",
    "AccessControlPolicy": {
      "AccessControlList": {
        "Grant": {
          "Grantee": {
            "xsi:type": "CanonicalUser",
            "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
            "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
          },
          "Permission": "FULL_CONTROL"
        }
      },
      "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
      "Owner": {
        "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
      }
    },
    "host": [
      "s3.us-west-2.amazonaws.com"
    ],
    "acl": [
      ""
    ]
  },
},

```

```

"responseElements": null,
"additionalEventData": {
  "SignatureVersion": "SigV4",
  "CipherSuite": "ECDHE-RSA-AES128-SHA",
  "AuthenticationMethod": "AuthHeader"
},
"requestID": "BD8798EACDD16751",
"eventID": "607b9532-1423-41c7-b048-ec2641693c47",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "s3.amazonaws.com"
}
},
{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/myUserName",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "myUserName"
  },
  "eventTime": "2019-02-01T03:26:37Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "GetBucketVersioning",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "",
  "userAgent": "[]",
  "requestParameters": {
    "host": [
      "s3.us-west-2.amazonaws.com"
    ],
    "bucketName": "amzn-s3-demo-bucket1",
    "versioning": [
      ""
    ]
  }
},
"responseElements": null,
"additionalEventData": {
  "SignatureVersion": "SigV4",

```

```

        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "AuthenticationMethod": "AuthHeader"
    },
    "requestID": "07D681279BD94AED",
    "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "s3.amazonaws.com"
    }
}
]
}

```

Beispiel: Amazon S3 in Outposts-Logdateieinträgen

Verwaltungsereignisse für Amazon S3 on Outposts sind verfügbar über AWS CloudTrail. Weitere Informationen finden Sie unter [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#).

Darüber hinaus können Sie optional die Protokollierung von Datenereignissen in [aktivieren AWS CloudTrail](#).

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen S3-Bucket in einer Region übermittelt werden. CloudTrail Logs für Ihre Outposts-Buckets enthalten ein neues Feld, das den Outpost identifiziert `edgeDeviceDetails`, in dem sich der angegebene Bucket befindet.

Zusätzliche Protokollfelder enthalten die angeforderte Aktion, Datum und Uhrzeit der Aktion sowie die Anforderungsparameter. CloudTrail Bei Protokolldateien handelt es sich nicht um einen geordneten Stack-Trace der öffentlichen API Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der eine [PutObject](#)Aktion für `demonstrierts3-outposts`.

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/yourUserName",

```

```

    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "yourUserName"
  },
  "eventTime": "2020-11-30T15:44:33Z",
  "eventSource": "s3-outposts.amazonaws.com",
  "eventName": "PutObject",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "26.29.66.20",
  "userAgent": "aws-cli/1.18.39 Python/3.4.10 Darwin/18.7.0 boto3/1.15.39",
  "requestParameters": {
    "expires": "Wed, 21 Oct 2020 07:28:00 GMT",
    "Content-Language": "english",
    "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
    "ObjectCannedACL": "BucketOwnerFullControl",
    "x-amz-server-side-encryption": "Aes256",
    "Content-Encoding": "gzip",
    "Content-Length": "10",
    "Cache-Control": "no-cache",
    "Content-Type": "text/html; charset=UTF-8",
    "Content-Disposition": "attachment",
    "Content-MD5": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
    "x-amz-storage-class": "Outposts",
    "x-amz-server-side-encryption-customer-algorithm": "Aes256",
    "bucketName": "amzn-s3-demo-bucket1",
    "Key": "path/upload.sh"
  },
  "responseElements": {
    "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
    "x-amz-server-side-encryption": "Aes256",
    "x-amz-version-id": "001",
    "x-amz-server-side-encryption-customer-algorithm": "Aes256",
    "ETag": "d41d8cd98f00b204e9800998ecf8427f"
  },
  "additionalEventData": {
    "CipherSuite": "ECDHE-RSA-AES128-SHA",
    "bytesTransferredIn": 10,
    "x-amz-id-2": "29xXQBv20
+x0HKItvzY1suLv1i6A52E0z0X159fpfsItYd58JhXwKxXAXI4IQkp6",
    "SignatureVersion": "SigV4",
    "bytesTransferredOut": 20,
    "AuthenticationMethod": "AuthHeader"
  }

```

```

    },
    "requestID": "8E96D972160306FA",
    "eventID": "ee3b4e0c-ab12-459b-9998-0a5a6f2e4015",
    "readOnly": false,
    "resources": [
      {
        "accountId": "222222222222",
        "type": "AWS::S3Outposts::Object",
        "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/path/upload.sh"
      },
      {
        "accountId": "222222222222",
        "type": "AWS::S3Outposts::Bucket",
        "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "444455556666",
    "sharedEventID": "02759a4c-c040-4758-b84b-7cbaaf17747a",
    "edgeDeviceDetails": {
      "type": "outposts",
      "deviceId": "op-01ac5d28a6a232904"
    },
    "eventCategory": "Data"
  }
}

```

Aktivieren der CloudTrail Ereignisprotokollierung für S3-Buckets und - Objekte

Sie können CloudTrail Datenereignisse verwenden, um Informationen über Anfragen auf Bucket- und Objektebene in Amazon S3 abzurufen. Um CloudTrail Datenereignisse für alle Ihre Buckets oder für eine Liste bestimmter Buckets zu aktivieren, müssen Sie manuell [einen Trail in erstellen](#). CloudTrail

Note

- Die Standardeinstellung für CloudTrail ist, dass nur Verwaltungsereignisse gesucht werden. Prüfen Sie, ob Datenereignisse für das Konto aktiviert wurden.

- Ein S3-Bucket mit hoher Workload kann in kurzer Zeit Tausende Protokolle generieren. Denken Sie daran, wie lange Sie CloudTrail Datenereignisse für einen stark frequentierten Bereich aktivieren möchten.

CloudTrail speichert Amazon S3 S3-Datenereignisprotokolle in einem S3-Bucket Ihrer Wahl. Erwägen Sie, einen Bucket in einem separaten AWS-Konto um Ereignisse aus mehreren Buckets, die Sie möglicherweise besitzen, besser an einem zentralen Ort zu organisieren, um Abfragen und Analysen zu vereinfachen. AWS Organizations hilft Ihnen bei der Erstellung eines AWS-Konto das ist mit dem Konto verknüpft, dem der Bucket gehört, den Sie überwachen. Weitere Informationen finden Sie unter [Was ist AWS Organizations?](#) in der AWS Organizations Benutzerleitfaden.

Wenn Sie Datenereignisse für einen Trail protokollieren, können Sie wählen CloudTrail, ob Sie erweiterte oder einfache Event-Selektoren verwenden möchten, um Datenereignisse für Objekte zu protokollieren, die in Allzweck-Buckets gespeichert sind. Um Datenereignisse für Objekte zu protokollieren, die in Verzeichnis-Buckets gespeichert sind, müssen Sie erweiterte Ereignisselectoren verwenden. Weitere Informationen finden Sie unter [Protokollieren mit AWS CloudTrail für S3 Express One Zone](#).

Wenn Sie in der CloudTrail Konsole mithilfe erweiterter Event-Selektoren einen Trail erstellen, können Sie im Bereich Datenereignisse die Option Alle Ereignisse protokollieren für die Protokollauswahl-Vorlage auswählen, um alle Ereignisse auf Objektebene zu protokollieren. Wenn Sie in der CloudTrail Konsole mithilfe einfacher Ereignisauswahlen einen Trail erstellen, können Sie im Bereich Datenereignisse das Kontrollkästchen Alle S3-Buckets in Ihrem Konto auswählen aktivieren, um alle Ereignisse auf Objektebene zu protokollieren.

Note

- Es hat sich bewährt, eine Lebenszykluskonfiguration für Ihre zu erstellen AWS CloudTrail Bucket für Datenereignisse. Konfigurieren Sie die Lebenszykluskonfiguration zum regelmäßigen Entfernen von Protokolldateien nach dem Zeitraum, der Ihres Erachtens für die Überprüfung erforderlich ist. Dadurch wird die Menge der Daten reduziert, die Athena in einer Abfrage analysiert. Weitere Informationen finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#).
- Weitere Informationen zum Format der Protokolle finden Sie unter [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#).

- Beispiele für das Abfragen von CloudTrail Logs finden Sie im AWS Big-Data-Blogbeitrag [Analysieren Sie Sicherheit, Compliance und betriebliche Aktivitäten mithilfe von AWS CloudTrail und Amazon Athena](#).

Aktivieren der Protokollierung für Objekte in einem Bucket mit der Konsole

Sie können die Amazon S3 S3-Konsole verwenden, um eine zu konfigurieren AWS CloudTrail Pfad zum Protokollieren von Datenereignissen für Objekte in einem S3-Bucket. CloudTrail unterstützt die Protokollierung von Amazon S3 API S3-Vorgängen auf Objektebene wie `GetObjectDeleteObject`, und `PutObject`. Diese Ereignisse werden als Datenereignisse bezeichnet.

Standardmäßig protokollieren CloudTrail Trails keine Datenereignisse, aber Sie können Trails so konfigurieren, dass Datenereignisse für von Ihnen angegebene S3-Buckets oder Datenereignisse für alle Amazon S3 S3-Buckets in Ihrem AWS-Konto. Weitere Informationen finden Sie unter [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#).

CloudTrail füllt keine Datenereignisse in der CloudTrail Ereignishistorie auf. Darüber hinaus werden nicht alle Aktionen auf Bucket-Ebene in den CloudTrail Ereignisverlauf eingetragen. Weitere Informationen zu den API Aktionen auf Amazon S3 S3-Bucket-Ebene, die durch CloudTrail Protokollierung verfolgt werden, finden Sie unter [Amazon S3 S3-Aktionen auf Bucket-Ebene, die durch Protokollierung verfolgt werden CloudTrail](#). Weitere Informationen zur Abfrage von CloudTrail Protokollen finden Sie im AWS Knowledge Center-Artikel über die [Verwendung von Amazon CloudWatch Logs-Filtermustern und Amazon Athena zum Abfragen von CloudTrail Protokollen](#).

Um einen Trail zum Protokollieren von Datenereignissen für einen S3-Bucket zu konfigurieren, können Sie entweder AWS CloudTrail Konsole oder die Amazon S3 S3-Konsole. Wenn Sie einen Trail konfigurieren, um Datenereignisse für alle Amazon S3 S3-Buckets in Ihrem zu protokollieren AWS-Konto, es ist einfacher, die CloudTrail Konsole zu verwenden. Informationen zur Verwendung der CloudTrail Konsole zur Konfiguration eines Trails zur Protokollierung von S3-Datenereignissen finden Sie unter [Datenereignisse](#) in der AWS CloudTrail Benutzerleitfaden.

Important


Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen finden Sie unter [AWS CloudTrail Preisgestaltung](#).

Das folgende Verfahren zeigt, wie Sie mit der Amazon S3 S3-Konsole einen CloudTrail Trail zum Protokollieren von Datenereignissen für einen S3-Bucket konfigurieren.

Um die Protokollierung von CloudTrail Datenereignissen für Objekte in einem S3-Allzweck-Bucket oder in einem S3-Verzeichnis-Bucket zu aktivieren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus.
3. Wählen Sie Properties (Eigenschaften).
4. Unter AWS CloudTrail Datenereignisse, wählen Sie Konfigurieren in CloudTrail.

Sie können einen neuen CloudTrail Trail erstellen oder einen vorhandenen Trail wiederverwenden und Amazon S3 S3-Datenereignisse so konfigurieren, dass sie in Ihrem Trail protokolliert werden. Informationen zum Erstellen von Trails in der CloudTrail Konsole finden Sie unter [Erstellen und Aktualisieren eines Trails mit der Konsole](#) in der AWS CloudTrail Benutzerleitfaden. Informationen zur Konfiguration der Amazon S3 S3-Datenereignisprotokollierung in der CloudTrail Konsole finden Sie unter [Protokollieren von Datenereignissen für Amazon S3 S3-Objekte](#) in der AWS CloudTrail Benutzerleitfaden.

 Note

Wenn Sie die CloudTrail Konsole oder die Amazon S3 S3-Konsole verwenden, um einen Trail zum Protokollieren von Datenereignissen für einen S3-Bucket zu konfigurieren, zeigt die Amazon S3 S3-Konsole an, dass die Protokollierung auf Objektebene für den Bucket aktiviert ist.

Um die Protokollierung CloudTrail von Datenereignissen für Objekte in einem S3-Bucket zu deaktivieren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die CloudTrail Konsole unter <https://console.aws.amazon.com/cloudtrail/>.
2. Wählen Sie im linken Navigationsbereich Trails aus.
3. Wählen Sie den Namen des Trails aus, den Sie erstellt haben, um Ereignisse für den Bucket zu protokollieren.
4. Wählen Sie oben rechts auf der Detailseite des Trails Protokollierung beenden aus.

5. Wählen Sie im anschließend angezeigten Dialogfeld Protokollierung beenden aus.

Weitere Informationen zum Aktivieren der Protokollierung auf Objektebene, wenn Sie einen S3-Bucket erstellen, finden Sie unter [Erstellen eines Buckets](#).

Weitere Informationen zur CloudTrail Protokollierung mit S3-Buckets finden Sie in den folgenden Themen:

- [Anzeigen der Eigenschaften eines S3-Buckets](#)
- [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#)
- [Arbeiten mit CloudTrail Protokolldateien](#) in der AWS CloudTrail Benutzerleitfaden

Identifizieren von Amazon S3 S3-Anfragen mit CloudTrail

In Amazon S3 können Sie Anfragen anhand eines identifizieren AWS CloudTrail Ereignisprotokoll. AWS CloudTrail ist die bevorzugte Methode zur Identifizierung von Amazon S3 S3-Anfragen. Wenn Sie jedoch Amazon S3 S3-Serverzugriffsprotokolle verwenden, finden Sie weitere Informationen unter [the section called "Identifizieren von S3-Anfragen"](#).

Themen

- [Identifizieren von Anfragen an Amazon S3 in einem CloudTrail Protokoll](#)
- [Identifizieren von Amazon S3 Signature Version 2-Anfragen mithilfe von CloudTrail](#)
- [Identifizieren des Zugriffs auf S3-Objekte mithilfe von CloudTrail](#)

Identifizieren von Anfragen an Amazon S3 in einem CloudTrail Protokoll

Nachdem Sie die Bereitstellung von Ereignissen CloudTrail an einen Bucket eingerichtet haben, sollten Sie auf der Amazon S3 S3-Konsole sehen, dass Objekte an Ihren Ziel-Bucket gesendet werden. Diese sind folgendermaßen formatiert:

```
s3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd
```

Von protokollierte Ereignisse CloudTrail werden komprimiert gespeichert, gzipped JSONObjekte in Ihrem S3-Bucket. Um Anfragen effizient zu finden, sollten Sie einen Dienst wie Amazon Athena verwenden, um die CloudTrail Protokolle zu indizieren und abzufragen.

Weitere Informationen zu CloudTrail und Athena finden Sie unter Tabelle [erstellen für AWS CloudTrail loggt sich in Athena mithilfe der Partitionsprojektion](#) im Amazon Athena Athena-Benutzerhandbuch ein.

Identifizieren von Amazon S3 Signature Version 2-Anfragen mithilfe von CloudTrail

Sie können ein CloudTrail Ereignisprotokoll verwenden, um zu ermitteln, welche API Signaturversion zum Signieren einer Anfrage in Amazon S3 verwendet wurde. Diese Funktion ist wichtig, weil die Unterstützung für Signature Version 2 deaktiviert wird (veraltet). Danach akzeptiert Amazon S3 keine Anforderungen mit Signature Version 2 mehr, alle Anforderungen müssen also mit Signature Version 4 signiert werden.

Wir empfehlen dringend, dass Sie es verwenden CloudTrail , um festzustellen, ob in einem Ihrer Workflows Signature Version 2-Signaturen verwendet werden. Korrigieren Sie diese Workflows, indem Sie die betreffenden Bibliotheken und den Code so aktualisieren, dass mittels Signature Version 4 signiert wird. Dadurch können Beeinträchtigungen der geschäftlichen Prozesse werden.

Weitere Informationen finden Sie unter [Ankündigung: AWS CloudTrail für Amazon S3 fügt neue Felder für erweiterte Sicherheitsüberprüfungen](#) hinzu AWS re:Post.

Note

CloudTrail Ereignisse für Amazon S3 enthalten die Signaturversion in den Anforderungsdetails unter dem Schlüsselnamen 'additionalEventData'. Um die Signaturversion von Anfragen für Objekte in Amazon S3 wie GET, PUT und DELETE Anfragen zu finden, müssen Sie CloudTrail Datenereignisse aktivieren. (Diese Option ist standardmäßig deaktiviert.)

AWS CloudTrail ist die bevorzugte Methode zur Identifizierung von Signature Version 2-Anfragen. Wenn Sie Zugriffsprotokolle von Amazon S3 Server verwenden, lesen Sie [Identifizieren von Signature-Version-2-Anforderungen mittels Amazon-S3-Zugriffsprotokollen](#).

Themen

- [Athena-Abfragebeispiele zum Identifizieren von Anfragen zur Amazon S3 Signature Version 2](#)
- [Partitionieren von Signature Version 2-Daten](#)

Athena-Abfragebeispiele zum Identifizieren von Anforderungen zur Amazon S3 Signature Version 2

Example – Wählen Sie alle Signature-Version-2-Ereignisse aus und drucken Sie nur **EventTime**, **S3_Action**, **Request_Parameters**, **Region**, **SourceIP** und **UserAgent**

Ersetzen Sie in der folgenden Athena-Abfrage *s3_cloudtrail_events_db.cloudtrail_table* mit Ihren Athena-Daten und erhöhen oder entfernen Sie das Limit nach Bedarf.

```
SELECT EventTime, EventName as S3_Action, requestParameters as Request_Parameters,
       awsregion as AWS_Region, sourceipaddress as Source_IP, useragent as User_Agent
FROM s3_cloudtrail_events_db.cloudtrail_table
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
LIMIT 10;
```

Example – Alle Anforderer auswählen, die Signature Version 2-Datenverkehr senden

```
SELECT useridentity.arn, Count(requestid) as RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table
WHERE eventsource='s3.amazonaws.com'
      and json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
Group by useridentity.arn
```

Partitionieren von Signature Version 2-Daten

Wenn Sie über eine große Datenmenge verfügen, die abgefragt werden muss, können Sie Kosten und Laufzeit für Athena reduzieren, indem Sie eine partitionierte Tabelle erstellen.

Erstellen Sie zu diesem Zweck folgendermaßen eine neue Tabelle mit Partitionen.

```
CREATE EXTERNAL TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned(
  eventversion STRING,
  userIdentity STRUCT<
    type:STRING,
    principalid:STRING,
    arn:STRING,
    accountid:STRING,
```

```

    invokedby:STRING,
    accesskeyid:STRING,
    userName:STRING,
    sessioncontext:STRUCT<
        attributes:STRUCT<
            mfaauthenticated:STRING,
            creationdate:STRING>,
            sessionIssuer:STRUCT<
                type:STRING,
                principalId:STRING,
                arn:STRING,
                accountId:STRING,
                userName:STRING>
        >
    >,
    eventTime STRING,
    eventSource STRING,
    eventName STRING,
    awsRegion STRING,
    sourceIpAddress STRING,
    userAgent STRING,
    errorCode STRING,
    errorMessage STRING,
    requestParameters STRING,
    responseElements STRING,
    additionalEventData STRING,
    requestId STRING,
    eventId STRING,
    resources ARRAY<STRUCT<ARN:STRING,accountId: STRING,type:STRING>>,
    eventType STRING,
    apiVersion STRING,
    readOnly STRING,
    recipientAccountId STRING,
    serviceEventDetails STRING,
    sharedEventID STRING,
    vpcEndpointId STRING
)
PARTITIONED BY (region string, year string, month string, day string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/';

```

Erstellen Sie dann die einzelnen Partitionen. Sie können keine Resultate aus noch nicht erstellten Daten ermitteln.

```
ALTER TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned ADD
  PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
  PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
  PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
  PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
  PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
  PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
  PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
  PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION
  's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/sa-east-1/2019/02/19/';
```

Anschließend können Sie die Anforderung basierend auf diesen Partitionen erstellen und müssen nicht mehr den gesamten Bucket laden.

```
SELECT useridentity.arn,
Count(requestid) AS RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table_partitioned
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
AND region='us-east-1'
AND year='2019'
AND month='02'
AND day='19'
Group by useridentity.arn
```

Identifizieren des Zugriffs auf S3-Objekte mithilfe von CloudTrail

Sie können Ihre verwenden AWS CloudTrail Ereignisprotokolle, um Amazon S3 S3-Objektzugriffsanforderungen für Datenereignisse wie `GetObject` `DeleteObject` `PutObject`, und zu identifizieren und zusätzliche Informationen zu diesen Anfragen zu ermitteln.

Das folgende Beispiel zeigt, wie Sie alle PUT Objektanfragen für Amazon S3 von einem abrufen AWS CloudTrail Ereignisprotokoll.

Themen

- [Athena-Abfragebeispiele zur Identifizierung von Amazon-S3-Objektzugriffsanfragen](#)

Athena-Abfragebeispiele zur Identifizierung von Amazon-S3-Objektzugriffsanfragen

Ersetzen Sie in den folgenden Athena-Abfragebeispielen

`s3_cloudtrail_events_db.cloudtrail_table` mit Ihren Athena-Daten und ändern Sie den Datumsbereich nach Bedarf.

Example – Wählen Sie alle Ereignisse aus, für die **PUT**-Objektzugriffsanforderungen vorliegen, und drucken Sie nur **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** und **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'PutObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example – Wählen Sie alle Ereignisse aus, für die **GET**-Objektzugriffsanforderungen vorliegen, und drucken Sie nur **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** und **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'GetObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example – Wählen Sie alle anonymen Anforderungseignisse für einen Bucket in einem bestimmten Zeitraum aus und drucken Sie nur **EventTime**, **EventName**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **UserARN** und **AccountID**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  userIdentity.arn as userArn,
  userIdentity.accountId
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  userIdentity.accountId = 'anonymous'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — Identifizieren Sie alle Anfragen, ACL für die eine Autorisierung erforderlich war

Das folgende Amazon Athena Athena-Abfragebeispiel zeigt, wie Sie alle Anfragen an Ihre S3-Buckets identifizieren können, für die eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich

war. Wenn ACL für die Anfrage eine Autorisierung erforderlich war, ist der `aclRequired` Wert in `additionalEventData`. Yes Wenn keine erforderlich ACLs waren, `aclRequired` ist er nicht vorhanden. Sie können diese Informationen verwenden, um diese ACL Berechtigungen auf die entsprechenden Bucket-Richtlinien zu migrieren. Nachdem Sie diese Bucket-Richtlinien erstellt haben, können Sie sie ACLs für diese Buckets deaktivieren. Weitere Informationen zur Deaktivierung ACLs finden Sie unter [Voraussetzungen für die Deaktivierung ACLs](#)

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  userIdentity.arn as userArn,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  json_extract_scalar(additionalEventData, '$.aclRequired') as aclRequired
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  json_extract_scalar(additionalEventData, '$.aclRequired') = 'Yes'
  AND eventTime BETWEEN '2022-05-10T00:00:00Z' and '2022-08-10T00:00:00Z'
```

Note

- Diese Abfragebeispiele können auch für die Sicherheitsüberwachung nützlich sein. Sie können die Ergebnisse auf PutObject- oder GetObject-Aufrufe von unerwarteten oder nicht autorisierten IP-Adressen oder Anforderern und zum Identifizieren anonymer Anforderungen an Ihre Buckets prüfen.
- Diese Abfrage ruft nur Informationen von der Zeit ab, zu der die Protokollierung aktiviert wurde.

Wenn Sie Amazon-S3-Server-Zugriffsprotokolle verwenden, lesen Sie [Identifizieren von Objektzugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen](#).

Protokollieren von Anfragen mit Server-Zugriffsprotokollierung

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anforderungen, die an einen Bucket gestellt wurden. Server-Zugriffsprotokolle sind für viele Anwendungen nützlich. Beispielsweise können Zugriffsprotokoll-Informationen bei Sicherheits- und Zugriffsprüfungen nützlich sein. Diese Informationen können Ihnen auch helfen, mehr über Ihre Kundenbasis zu erfahren und Ihre Amazon-S3-Rechnung zu verstehen.

Note

Serverzugriffsprotokolle protokollieren keine Informationen zu Fehlern wegen Umleitungen in falsche Regionen, die nach dem 20. März 2019 gestartet wurden. Fehler wegen Umleitungen in falsche Regionen treten auf, wenn eine Anforderung für ein Objekt oder einen Bucket außerhalb der Region gesendet wird, in der sich der Bucket befindet.

Wie aktiviere ich die Protokollzustellung?

Um die Protokollbereitstellung zu aktivieren, führen Sie die folgenden grundlegenden Schritte aus. Details hierzu finden Sie unter [Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung](#).

1. Geben Sie den Namen des Ziel-Buckets (auch Ziel-Bucket genannt) an. In diesem Bucket soll Amazon S3 die Zugriffsprotokolle als Objekte speichern. Sowohl der Quell- als auch der Ziel-Bucket müssen sich im selben Bereich befinden AWS-Region und gehören demselben Konto. Der Ziel-Bucket darf keine S3-Object-Lock-Standardkonfiguration für die Aufbewahrungsdauer besitzen. Für den Ziel-Bucket darf außerdem die Option „Zahlung durch den Anforderer“ nicht aktiviert sein.

Sie können Protokolle in jeden Bucket speichern lassen, der sich in der gleichen Region wie der Quell-Bucket befindet, einschließlich des Quell-Buckets selbst. Zur einfacheren Protokollverwaltung empfehlen wir jedoch, Zugriffsprotokolle in einem anderen Bucket zu speichern.

Wenn Quell- und Ziel-Bucket identisch sind, werden zusätzliche Protokolle für die Protokolle erstellt, die zum Bucket geschrieben werden. Hierdurch entsteht eine Endlosprotokollschleife. Diese Vorgehensweise wird nicht empfohlen, da sie zu einer geringfügigen Erhöhung der Speicherkosten führen könnte. Weiterhin könnten die zusätzlichen Protokolle über Protokolle das Auffinden des gesuchten Protokolls erschweren.

Wenn Sie Zugriffsprotokolle im Quell-Bucket speichern möchten, sollten Sie ein Zielpräfix (Zielpräfix) für alle Protokollobjektschlüssel angeben. Wenn Sie ein Präfix angeben, beginnen alle Protokollobjektnamen mit einer gemeinsamen Zeichenfolge. So können Sie die Protokollobjekte leichter identifizieren.

- (Optional) Weisen Sie allen Amazon-S3-Protokollobjektschlüsseln ein Zielpräfix zu. Das Zielpräfix (Zielpräfix) macht es einfacher für Sie, die Protokollobjekte zu finden. Wenn Sie beispielsweise den Präfixwert `logs/` angeben, beginnt jedes von Amazon S3 erstellte Protokollobjekt mit dem Präfix `logs/` im Schlüssel, z. B.:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

Wenn Sie den Präfixwert `logs` angeben, sieht das Protokollobjekt wie folgt aus:

```
logs2013-11-01-21-32-16-E568B2907131C0C0
```

[Präfixe](#) sind auch nützlich, um zwischen Quell-Buckets zu unterscheiden, wenn mehrere Buckets zum selben Ziel-Bucket protokolliert werden.

Dieses Präfix ist auch beim Löschen der Protokolle nützlich. Beispielsweise können Sie eine Lebenszyklus-Konfigurationsregel für Amazon S3 festlegen, um Objekte mit einem bestimmten Präfix zu löschen. Weitere Informationen finden Sie unter [Löschen von Amazon-S3-Protokolldateien](#).

- (Optional) Legen Sie Berechtigungen fest, um anderen Benutzern Zugriff auf die generierten Protokolle zu gewähren. Standardmäßig hat nur der Bucket-Eigentümer stets vollen Zugriff auf die Protokollobjekte. Wenn Ihr Ziel-Bucket die vom Bucket-Besitzer erzwungene Einstellung für S3 Object Ownership verwendet, um Zugriffskontrolllisten (ACLs) zu deaktivieren, können Sie keine Berechtigungen für Zielzuschüsse (auch als Zielzuschüsse bezeichnet) gewähren, die verwenden ACLs. Sie können jedoch Ihre Bucket-Richtlinie für den Ziel-Bucket aktualisieren, um anderen Benutzern Zugriff zu gewähren. Weitere Informationen erhalten Sie unter [Identity and Access Management für Amazon S3](#) und [Berechtigungen für Protokollbereitstellung](#).
- (Optional) Legen Sie ein Protokollobjekt-Schlüsselformat für die Protokolldateien fest. Es gibt zwei Optionen für das Protokollobjekt-Schlüsselformat (auch als Zielobjekt-Schlüsselformat bezeichnet):
 - on-date-based N-Partitionierung — Dies ist das ursprüngliche Schlüsselformat für Log-Objekte. Wenn Sie dieses Format wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Wenn Sie beispielsweise `logs/` als Präfix angeben, werden Ihre Protokollobjekte wie folgt benannt:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

- Partitionierung mit Datum – Wenn Sie die Partitionierung mit Datum wählen, können Sie die Ereignis- oder Bereitstellungszeit für die Protokolldatei als die Datumsquelle wählen, die im Protokollformat verwendet wird. Dieses Format vereinfacht das Abfragen der Protokolle.

Wenn Sie die Partitionierung mit Datum wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Wenn Sie beispielsweise `logs/` als Zielpräfix angeben, werden Ihre Protokollobjekte wie folgt benannt:

```
logs/123456789012/us-west-2/DOC-EXAMPLE-SOURCE-  
BUCKET/2023/03/01/2023-03-01-21-32-16-E568B2907131C0C0
```

Bei der Bereitstellung nach Bereitstellungszeit entspricht die Zeit in den Namen der Protokolldateien der Bereitstellungszeit für die Protokolldateien.

Bei der Bereitstellung nach Ereigniszeit entsprechen die Angaben für Jahr, Monat und Tag dem Tag, an dem das Ereignis eingetreten ist. Stunde, Minuten und Sekunden sind im Schlüssel auf `00` festgelegt. Die in diesen Protokolldateien bereitgestellten Protokolle beziehen sich nur auf einen bestimmten Tag.

Wenn Sie Ihre Logs über das konfigurieren AWS Command Line Interface (AWS CLI), AWS SDKs, oder Amazon S3 RESTAPI, verwenden Sie, `TargetObjectKeyFormat` um das Schlüsselformat für das Protokollobjekt anzugeben. Um die non-date-based Partitionierung anzugeben, verwenden Sie `SimplePrefix`. Um eine Partitionierung mit Datum anzugeben, verwenden Sie `PartitionedPrefix`. Bei Verwendung von `PartitionedPrefix` verwenden Sie `PartitionDateSource`, um `EventTime` oder `DeliveryTime` anzugeben.

Für `SimplePrefix` sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[TargetPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Für `PartitionedPrefix` mit Ereignis- oder Bereitstellungszeit sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[TargetPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/  
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Protokollobjekt-Schlüsselformat

Amazon S3 verwendet die folgenden Objektschlüsselformate für die Protokollobjekte, die in den Ziel-Bucket hochgeladen werden:

- on-date-based N-Partitionierung — Dies ist das ursprüngliche Schlüsselformat für Protokollobjekte. Wenn Sie dieses Format wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

- Partitionierung mit Datum – Wenn Sie die Partitionierung mit Datum wählen, können Sie die Ereignis- oder Bereitstellungszeit für die Protokolldatei als die Datumsquelle wählen, die im Protokollformat verwendet wird. Dieses Format vereinfacht das Abfragen der Protokolle.

Wenn Sie die Partitionierung mit Datum wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/  
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Im Protokollobjektschlüssel sind YYYY, MM, DD, hh, mm und ss jeweils die Stellen für Jahr, Monat, Tag, Stunde, Minute und Sekunden. Diese Daten und Uhrzeiten sind in koordinierter Weltzeit (UTC) angegeben.

Eine Protokolldatei, die zu einem bestimmten Zeitpunkt bereitgestellt wurde, kann Datensätze enthalten, die an einem beliebigen Zeitpunkt davor geschrieben wurden. Es lässt sich nicht

feststellen, ob alle Protokoll-Datensätze für ein bestimmtes Zeitintervall bereitgestellt wurden oder nicht.

Die `UniqueString`-Komponente des Schlüssels verhindert, dass Dateien überschrieben werden. Sie hat keine Bedeutung und wird normalerweise von Protokollverarbeitungssoftware ignoriert.

Wie werden Protokolle ausgeliefert?

Amazon S3 sammelt regelmäßig Zugriffprotokoll-Datensätze, konsolidiert die Datensätze in Protokolldateien und lädt anschließend die Protokolldateien als Protokollobjekte zu Ihrem Ziel-Bucket hoch. Wenn Sie die Protokollierung für mehrere Quell-Buckets mit demselben Ziel-Bucket aktivieren, besitzt der Ziel-Bucket Zugriffprotokolle für alle diese Quell-Buckets. Jedes Protokollobjekt gibt jedoch Zugriffprotokoll-Datensätze für einen bestimmten Quell-Bucket aus.

Amazon S3 verwendet ein spezielles Protokollbereitstellungskonto zum Schreiben der Server-Zugriffprotokolle. Für diese Protokolle gelten die normalen Zugriffskontrollbeschränkungen. Wir empfehlen Ihnen, die Bucket-Richtlinie für den Ziel-Bucket zu aktualisieren, um Zugriff auf den Protokollservice-Prinzipal (`logging.s3.amazonaws.com`) zur Zugriffprotokollbereitstellung zu gewähren. Sie können der S3-Protokollzustellungsgruppe auch über Ihre Bucket-Zugriffskontrollliste (ACL) Zugriff auf die Übermittlung von Zugriffprotokollen gewähren. Es wird jedoch nicht empfohlen, mithilfe Ihres Buckets ACL Zugriff auf die S3-Protokollbereitstellungsgruppe zu gewähren.

Wenn Sie die Serverzugriffprotokollierung über die Ziel-Bucket-Richtlinie aktivieren und Zugriff für die Zugriffprotokollbereitstellung gewähren, müssen Sie die Richtlinie aktualisieren, um `s3:PutObject`-Zugriff für den Protokollierungsservice-Prinzipal zuzulassen. Wenn Sie die Serverzugriffprotokollierung über die Amazon-S3-Konsole aktivieren, aktualisiert die Konsole die Ziel-Bucket-Richtlinie automatisch, um dem Protokollierungsservice-Prinzipal diese Berechtigungen zu gewähren. Weitere Informationen zum Erteilen von Berechtigungen für die Zustellung von Serverzugriffprotokollen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Note

S3 unterstützt nicht die Übermittlung von CloudTrail Protokollen oder Serverzugriffprotokollen an den Anforderer oder den Bucket-Besitzer für VPC Endpunktanforderungen, wenn die VPC Endpunktrichtlinie sie ablehnt, oder für Anfragen, die fehlschlagen, bevor die VPC Richtlinie ausgewertet wurde.

Vom Bucket-Eigentümer erzwungene Einstellung für S3 Object Ownership

Wenn der Ziel-Bucket die erzwungene Einstellung des Bucket-Besitzers für Objekteigentum verwendet, ACLs sind sie deaktiviert und wirken sich nicht mehr auf die Berechtigungen aus. Sie müssen die Bucket-Richtlinie aktualisieren, damit der Ziel-Bucket Zugriff auf den Protokollierungsservice-Prinzipal gewähren kann. Informationen zu Object Ownership finden Sie unter [Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs](#).

Best-Effort-Serverprotokollbereitstellung

Serverzugriffsprotokoll-Datensätze werden auf Best-Effort-Grundlage bereitgestellt. Die meisten Anforderungen nach einem Bucket, der für die Protokollierung richtig konfiguriert ist, führen zu einem ausgelieferten Protokollsatz. Die meisten Protokollsätze werden innerhalb weniger Stunden nach der Aufnahme geliefert, können aber häufiger geliefert werden.

Die Vollständigkeit und Aktualität der Serverprotokollierung wird nicht garantiert. Der Protokolldatensatz für eine bestimmte Anforderung wird möglicherweise viel später bereitgestellt, als die Anforderung tatsächlich verarbeitet wurde; es kann auch sein, dass er gar nicht bereitgestellt wird. Es ist möglich, dass Sie sogar ein Duplikat eines Protokolldatensatzes sehen. Der Zweck der Serverprotokolle besteht darin, Ihnen einen Überblick über die Art des Datenverkehrs zu und von Ihrem Bucket zu vermitteln. Auch wenn Protokolldatensätze selten verloren gehen oder dupliziert werden, sollten Sie wissen, dass die Serverprotokollierung nicht der vollständigen Erfassung aller Anforderungen dient.

Aufgrund der Best-Effort-Grundlage der Serverprotokollierung enthalten Ihre Nutzungsberichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in einem bereitgestellten Serverprotokoll angezeigt werden. Sie finden diese Nutzungsberichte unter Kosten- und Nutzungsberichte im AWS Billing and Cost Management console.

Statusänderungen in der Bucket-Protokollierung werden mit der Zeit wirksam

Änderungen am Protokollierungsstatus eines Buckets benötigen einige Zeit, bis sie sich auf die Bereitstellung von Protokolldateien auswirken. Wenn Sie die Protokollierung für einen Bucket aktivieren, werden in der folgenden Stunde möglicherweise einige Anforderungen protokolliert und andere nicht. Angenommen, Sie ändern den Ziel-Bucket für die Protokollierung von Bucket A in Bucket B. In der nächsten Stunde werden möglicherweise einige Protokolle in Bucket A bereitgestellt, während andere Protokolle im neuen Ziel-Bucket B bereitgestellt werden. In jedem Fall werden die neuen Einstellungen letztendlich ohne weiteres Eingreifen Ihrerseits wirksam.

Weitere Informationen zu Protokollierungs- und Protokolldateien finden Sie in den folgenden Abschnitten:

Themen

- [Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung](#)
- [Amazon-S3-Server-Zugriffsprotokollformat](#)
- [Löschen von Amazon-S3-Protokolldateien](#)
- [Verwenden von Amazon-S3-Serverzugriffsprotokollen zur Identifizierung von Anforderungen](#)

Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an einen Amazon-S3-Bucket gestellt wurden. Server-Zugriffsprotokolle sind für viele Anwendungen nützlich. Beispielsweise können Zugriffsprotokoll-Informationen bei Sicherheits- und Zugriffsprüfungen nützlich sein. Diese Informationen können Ihnen auch helfen, mehr über Ihre Kundenbasis zu erfahren und Ihre Amazon-S3-Rechnung zu verstehen.

Standardmäßig erfasst Amazon S3 keine Server-Zugriffsprotokolle. Wenn Sie die Protokollierung aktivieren, stellt Amazon S3 Zugriffsprotokolle für einen Quell-Bucket in einem von Ihnen ausgewählten Ziel-Bucket (Ziel-Bucket) bereit. Der Ziel-Bucket muss sich im selben Bucket befinden AWS-Region and AWS-Konto wie der Quell-Bucket.

Ein Zugriffsprotokollsatz enthält Details über die Anforderungen, die an einen Bucket gestellt werden. Dabei kann es sich um den Anforderungstyp, die in der Anfrage angegebenen Ressourcen sowie Uhrzeit und Datum der Anfrageverarbeitung handeln. Weitere Informationen zu den Grundlagen der Protokollierung finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

Important

- Für die Aktivierung der Server-Zugriffsprotokollierung auf einem Amazon-S3-Bucket fallen keine zusätzlichen Kosten an. Für die Speicherung der Protokolldateien, die das System an Sie überträgt, fallen allerdings die normalen Gebühren an. (Sie können die Protokolldateien jederzeit löschen.) Wir berechnen keine Datenübertragungskosten für die Bereitstellung von Protokolldateien. Für den Zugriff auf die Protokolldateien fallen jedoch die regulären Kosten für Datenübertragungen an.
- Für Ihren Ziel-Bucket sollte keine Serverzugriffsprotokollierung aktiviert sein. Sie können Protokolle in jeden Bucket speichern lassen, der sich in der gleichen Region wie der Quell-

Bucket befindet, einschließlich des Quell-Buckets selbst. Die Zustellung von Protokollen an den Quell-Bucket führt jedoch zu einer unendlichen Schleife von Protokollen und wird nicht empfohlen. Zur einfacheren Protokollverwaltung empfehlen wir, Zugriffsprotokolle in einem anderen Bucket zu speichern. Weitere Informationen finden Sie unter [Wie aktiviere ich die Protokollzustellung?](#)

- S3-Buckets, für die S3 Object Lock aktiviert ist, können nicht als Ziel-Buckets für Serverzugriffsprotokolle verwendet werden. Ihr Ziel-Bucket darf nicht für eine Standardaufbewahrungsdauer konfiguriert sein.
- Für den Ziel-Bucket darf die Option „Zahlung durch den Anforderer“ nicht aktiviert sein.
- Sie können die [Standard-Bucket-Verschlüsselung](#) für den Ziel-Bucket nur verwenden, wenn Sie serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwenden, die den 256-Bit-Advanced Encryption Standard (-256) verwenden. AES Serverseitige Standardverschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) werden nicht unterstützt.

Sie können die Serverzugriffsprotokollierung aktivieren oder deaktivieren, indem Sie die Amazon S3-Konsole verwenden, Amazon S3API, AWS Command Line Interface (AWS CLI), oder AWS SDKs.

Berechtigungen für Protokollbereitstellung

Amazon S3 verwendet ein spezielles Protokollbereitstellungskonto zum Schreiben der Server-Zugriffsprotokolle. Für diese Protokolle gelten die normalen Zugriffskontrollbeschränkungen. Zur Bereitstellung von Zugriffsprotokollen müssen Sie dem Protokollierungsservice-Prinzipal (`logging.s3.amazonaws.com`) Zugriff auf den Ziel-Bucket gewähren.

Um Amazon S3 Berechtigungen für die Protokollzustellung zu gewähren, können Sie entweder eine Bucket-Richtlinie oder Bucket-Zugriffskontrolllisten (ACLs) verwenden, abhängig von den Einstellungen für den Besitz von S3-Objekten in Ihrem Ziel-Bucket. Wir empfehlen jedoch, stattdessen eine Bucket-Richtlinie zu verwendenACLs.

Vom Bucket-Eigentümer erzwungene Einstellung für S3 Object Ownership

Wenn der Ziel-Bucket die erzwungene Einstellung für Object Ownership verwendet, ACLs sind sie deaktiviert und wirken sich nicht mehr auf die Berechtigungen aus. In diesem Fall müssen Sie die Bucket-Richtlinie für den Ziel-Bucket aktualisieren, um Zugriff auf den Protokollierungsservice-Prinzipal zu gewähren. Sie können Ihren Bucket nicht aktualisierenACL, um Zugriff auf die

S3-Protokollbereitstellungsgruppe zu gewähren. Sie können auch keine Zielzuschüsse (auch Zielzuschüsse genannt) in Ihre aufnehmen [PutBucketLogging](#)-Konfiguration.

Informationen zur Migration eines vorhandenen Buckets ACLs für die Zugriffs-Log-Zustellung zu einer Bucket-Richtlinie finden Sie unter [Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs](#). Informationen zu Object Ownership finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#). Wenn Sie neue Buckets erstellen, ACLs sind diese standardmäßig deaktiviert.

Gewähren von Zugriff über eine Bucket-Richtlinie

Um über die Bucket-Richtlinie Zugriff auf den Ziel-Bucket zu gewähren, aktualisieren Sie die Bucket-Richtlinie so, dass sie die `s3:PutObject`-Berechtigung für den Protokollierungsservice-Prinzipal zulässt. Wenn Sie die Serverzugriffsprotokollierung über die Amazon-S3-Konsole aktivieren, aktualisiert die Konsole die Bucket-Richtlinie für den Ziel-Bucket automatisch, um dem Protokollierungsservice-Prinzipal diese Berechtigung zu gewähren. Wenn Sie die Serverzugriffsprotokollierung programmgesteuert aktivieren, müssen Sie die Bucket-Richtlinie für den Ziel-Bucket manuell aktualisieren, um Zugriff auf den Protokollierungsservice-Prinzipal zu gewähren.

Ein Beispiel für eine Bucket-Richtlinie, die Zugriff auf den Protokollierungsservice-Prinzipal gewährt, finden Sie unter [the section called "Erteilen von Berechtigungen für den Prinzipal des Protokollierungsservices mithilfe einer Bucket-Richtlinie"](#).

Zugriff mithilfe eines Buckets gewähren ACLs

Sie können auch Bucket verwenden, ACLs um Zugriff auf die Zugriffs-Log-Zustellung zu gewähren. Sie fügen dem Bucket einen Grant-Eintrag hinzuACL, der der S3-Protokollbereitstellungsgruppe `READ_ACP` Berechtigungen gewährtWRITE. Es wird jedoch nicht empfohlen, mithilfe eines Buckets ACLs Zugriff auf die S3-Protokollbereitstellungsgruppe zu gewähren. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#). Hinweise zur Migration eines vorhandenen Buckets ACLs für die Zugriffs-Log-Zustellung zu einer Bucket-Richtlinie finden Sie unter [Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs](#). Ein BeispielACL, das Zugriff auf den Logging-Dienstprinzipal gewährt, finden Sie unter [the section called "Gewähren Sie der Protokollbereitstellungsgruppe mithilfe eines Buckets Berechtigungen ACL"](#).

Erteilen von Berechtigungen für den Prinzipal des Protokollierungsservices mithilfe einer Bucket-Richtlinie

Diese Bucket-Beispielrichtlinie gewährt dem Protokollierungsservice-Prinzipal (`logging.s3.amazonaws.com`) die `s3:PutObject`-Berechtigung. Wenn Sie diese Bucket-Richtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen. In der folgenden Richtlinie *amzn-s3-demo-destination-bucket* handelt es sich um den Ziel-Bucket, in den Serverzugriffsprotokolle übermittelt werden, und *amzn-s3-demo-source-bucket* um den Quell-Bucket. *EXAMPLE-LOGGING-PREFIX* ist das optionale Zielpräfix (auch als Zielpräfix bezeichnet), das Sie für Ihre Protokollobjekte verwenden möchten. *SOURCE-ACCOUNT-ID* ist der AWS-Konto dem der Quell-Bucket gehört.

Note

Wenn Ihre Bucket-Richtlinie Deny-Anweisungen enthält, stellen Sie sicher, dass diese Anweisungen Amazon S3 nicht daran hindern, Zugriffsprotokolle bereitzustellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3ServerAccessLogsPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "logging.s3.amazonaws.com"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/EXAMPLE-LOGGING-PREFIX",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
        },
        "StringEquals": {
          "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
        }
      }
    }
  ]
}
```

```
]
}
```

Gewähren Sie der Protokollbereitstellungsgruppe mithilfe eines Buckets Berechtigungen ACL

Note

Aus Sicherheitsgründen deaktiviert Amazon S3 die Zugriffskontrolllisten (ACLs) standardmäßig in allen neuen Buckets. Weitere Informationen zu ACL Berechtigungen in der Amazon S3 S3-Konsole finden Sie unter [Konfiguration ACLs](#).

Obwohl wir diesen Ansatz nicht empfehlen, können Sie der Protokollbereitstellungsgruppe mithilfe eines Buckets Berechtigungen erteilen ACL. Wenn der Ziel-Bucket jedoch die vom Bucket-Besitzer erzwungene Einstellung für Object Ownership verwendet, können Sie Bucket oder Objekt ACLs nicht festlegen. Sie können auch keine Zielzuschüsse (auch als Zielzuschüsse bezeichnet) in Ihre [PutBucketLogging](#)-Konfiguration. Stattdessen müssen Sie eine Bucket-Richtlinie verwenden, um dem Prinzipal des Protokollierungsservices (`logging.s3.amazonaws.com`) Zugriff zu gewähren. Weitere Informationen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Im Bucket ACL wird die Protokollbereitstellungsgruppe wie folgt dargestellt URL:

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Um Berechtigungen WRITE und READ_ACP (ACLLeese-) Berechtigungen zu erteilen, fügen Sie dem Ziel-Bucket die folgenden Berechtigungen hinzu ACL:

```
<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
  </Grantee>
  <Permission>WRITE</Permission>
</Grant>
<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
  </Grantee>
  <Permission>READ_ACP</Permission>
</Grant>
```

Beispiele für das programmgesteuerte Hinzufügen von ACL Zuschüssen finden Sie unter.

[Konfiguration ACLs](#)

Important

Wenn Sie die Protokollierung des Amazon S3 S3-Serverzugriffs aktivieren, indem Sie AWS CloudFormation für einen Bucket, den Sie verwenden ACLs, um Zugriff auf die S3-Protokollbereitstellungsgruppe `AccessControl`: `LogDeliveryWrite` zu gewähren, müssen Sie Ihrer CloudFormation Vorlage auch "hinzufügen. Dies ist wichtig, da Sie diese Berechtigungen nur gewähren können, indem Sie eine ACL für den Bucket erstellen, aber Sie können keine benutzerdefinierten Berechtigungen ACLs für Buckets in CloudFormation erstellen. Sie können nur in Dosen ACLs mit CloudFormation verwenden.

Vorgehensweise zum Aktivieren der Server-Zugriffsprotokollierung

Um die Serverzugriffsprotokollierung mithilfe der Amazon S3-Konsole, Amazon S3, zu aktivieren RESTAPI, AWS SDKs, und AWS CLI, verwenden Sie die folgenden Verfahren.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie die Server-Zugriffsprotokollierung aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie im Abschnitt Server access logging (Server-Zugriffsprotokollierung) die Option Edit (Bearbeiten) aus.
5. Wählen Sie unter Serverzugriffsprotokollierung) die Option Aktivieren aus.
6. Geben Sie unter Ziel-Bucket einen Bucket und ein optionales Präfix an. Wenn Sie ein Präfix angeben, sollten Sie nach dem Präfix einen Schrägstrich (/) einfügen, damit Sie Ihre Protokolle leichter finden können.

Note

Wenn Sie ein Präfix mit einem Schrägstrich (/) angeben, können Sie die Protokollobjekte leichter finden. Wenn Sie beispielsweise den Präfixwert `logs/` angeben, beginnt jedes von Amazon S3 erstellte Protokollobjekt mit dem Präfix `logs/` im Schlüssel wie folgt:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

Wenn Sie den Präfixwert `logs` angeben, sieht das Protokollobjekt wie folgt aus:

```
logs2013-11-01-21-32-16-E568B2907131C0C0
```

7. Führen Sie unter Protokollobjekt-Schlüsselformat eine der folgenden Aktionen aus:
 - Um die non-date-based Partitionierung auszuwählen, wählen Sie `[DestinationPrefixYYYY] [] - [MM] - [DD] - [hh] - [mm] - [ss] - [UniqueString]`.
 - Um die datumsbasierte Partitionierung zu wählen, wählen Sie `[DestinationPrefixSourceAccountId] [SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/[] - [MMYYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - [UniqueString]` und wählen Sie dann S3 event time oder Log file delivery time.
8. Wählen Sie Änderungen speichern.

Wenn Sie die Serverzugriffsprotokollierung für einen Bucket aktivieren, aktiviert die Konsole die Protokollierung für den Quell-Bucket und aktualisiert die Bucket-Richtlinie für den Ziel-Bucket, um dem Protokollierungsservice-Prinzipal (`logging.s3.amazonaws.com`) die `s3:PutObject`-Berechtigung zu gewähren. Weitere Informationen zu dieser Bucket-Richtlinie finden Sie unter [Erteilen von Berechtigungen für den Prinzipal des Protokollierungsservices mithilfe einer Bucket-Richtlinie](#).

Sie können die Protokolle im Ziel-Bucket anzeigen. Nachdem Sie die Server-Zugriffsprotokollierung aktiviert haben, kann es einige Stunden dauern, bis die Protokolle in den Ziel-Bucket geliefert werden. Weitere Informationen darüber, wie und wann Protokolle bereitgestellt werden, finden Sie unter [Wie werden Protokolle ausgeliefert?](#).

Weitere Informationen finden Sie unter [Anzeigen der Eigenschaften eines S3-Buckets](#).

Verwenden Sie REST API

Um die Protokollierung zu aktivieren, reichen Sie ein [PutBucketLogging](#)Anfrage zum Hinzufügen der Logging-Konfiguration zum Quell-Bucket. Die Anforderung gibt den Ziel-Bucket (Ziel-Bucket) und optional das Präfix an, das für alle Protokollobjektschlüssel verwendet werden soll.

Das folgende Beispiel identifiziert sich *amzn-s3-demo-destination-bucket* als Ziel-Bucket und *logs/* als Präfix.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Das folgende Beispiel identifiziert sich *amzn-s3-demo-destination-bucket* als Ziel-Bucket, *logs/* als Präfix und *EventTime* als Schlüsselformat für das Protokollobjekt.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
    <TargetObjectKeyFormat>
      <PartitionedPrefix>
        <PartitionDateSource>EventTime</PartitionDateSource>
      </PartitionedPrefix>
    </TargetObjectKeyFormat>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Die Protokollobjekte werden vom S3-Protokollübermittlungskonto geschrieben und besitzen dessen Eigentümer, und der Bucket-Eigentümer erhält vollständige Berechtigungen für die Protokollobjekte. Sie können optional Zielgewährungen verwenden (Zielgewährungen), um anderen Benutzern Berechtigungen zu gewähren, damit diese auf die Protokolle zugreifen können. Weitere Informationen finden Sie unter [PutBucketLogging](#).

Note

Wenn der Ziel-Bucket die Einstellung „Von Bucket-Besitzer erzwungen“ für Object Ownership verwendet, können Sie keine Zielgewährungen verwenden, um anderen Benutzern Berechtigungen zu gewähren. Um anderen Benutzern Berechtigungen zu gewähren, können Sie die Bucket-Richtlinie für den Ziel-Bucket aktualisieren. Weitere Informationen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Um die Logging-Konfiguration für einen Bucket abzurufen, verwenden Sie [GetBucketLogging](#) API-Operation.

Um die Protokollierungskonfiguration zu löschen, senden Sie eine `PutBucketLogging`-Anforderung mit einem leeren `BucketLoggingStatus`:

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
</BucketLoggingStatus>
```

Um die Protokollierung in einem Bucket zu aktivieren, können Sie entweder Amazon S3 API oder AWS SDK-Wrapper-Bibliotheken.

Verwendung der AWS SDKs

In den folgenden Beispielen wird die Protokollierung für einen Bucket aktiviert. Sie müssen zwei Buckets erstellen, einen Quell-Bucket und einen Ziel-Bucket. Die Beispiele aktualisieren zuerst den Bucket ACL im Ziel-Bucket. Anschließend werden der Protokollbereitstellungsgruppe die notwendigen Berechtigungen für das Schreiben von Protokollen zum Ziel-Bucket gewährt, bevor die Protokollierung für den Quell-Bucket aktiviert wird.

Diese Beispiele funktionieren nicht für Ziel-Buckets, die die Einstellung „Von Bucket-Besitzer erzwungen“ für Object Ownership verwenden.

Wenn der Ziel-Bucket (Ziel-Bucket) die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum verwendet, können Sie weder Bucket noch Objekt ACLs festlegen. Sie können auch keine Zielzuschüsse (Zielzuschüsse) in Ihre [PutBucketLogging](#)-Konfiguration aufnehmen. Sie müssen eine Bucket-Richtlinie verwenden, um dem Protokollierungsserviceprinzipal (`logging.s3.amazonaws.com`) Zugriff zu gewähren. Weitere Informationen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).


```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
/// this example. The first is the bucket for which you wish to enable
/// logging, and the second is the location where you want to store the
/// logs.
/// </summary>
public class ServerAccessLogging
{
    private static IConfiguration _configuration = null!;

    public static async Task Main()
    {
        LoadConfig();

        string bucketName = _configuration["BucketName"];
        string logBucketName = _configuration["LogBucketName"];
        string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
        string accountId = _configuration["AccountId"];

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
        IAmazonS3 client = new AmazonS3Client();

        try
        {
            // Update bucket policy for target bucket to allow delivery of
logs to it.
            await SetBucketPolicyToAllowLogDelivery(
                client,
                bucketName,
                logBucketName,
                logObjectKeyPrefix,
                accountId);
        }
    }
}
```

```
        // Enable logging on the source bucket.
        await EnableLoggingAsync(
            client,
            bucketName,
            logBucketName,
            logObjectKeyPrefix);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine($"Error: {e.Message}");
    }
}

/// <summary>
/// This method grants appropriate permissions for logging to the
/// Amazon S3 bucket where the logs will be stored.
/// </summary>
/// <param name="client">The initialized Amazon S3 client which will be
used
/// to apply the bucket policy.</param>
/// <param name="sourceBucketName">The name of the source bucket.</param>
/// <param name="logBucketName">The name of the bucket where logging
/// information will be stored.</param>
/// <param name="logPrefix">The logging prefix where the logs should be
delivered.</param>
/// <param name="accountId">The account id of the account where the
source bucket exists.</param>
/// <returns>Async task.</returns>
public static async Task SetBucketPolicyToAllowLogDelivery(
    IAmazonS3 client,
    string sourceBucketName,
    string logBucketName,
    string logPrefix,
    string accountId)
{
    var resourceArn = @$"arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @$""";

    var newPolicy = @"{
        ""Statement"": [{
            ""Sid"": ""S3ServerAccessLogsPolicy"",
            ""Effect"": ""Allow"",
```

```

        ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
        ""Action"": [""s3:PutObject""],
        ""Resource"": ["" + resourceArn + @""],
        ""Condition"": {
        ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @"""" },
        ""StringEquals"": { ""aws:SourceAccount"": """" +
accountId + @"""" }
        }
    }
}];

    Console.WriteLine($"The policy to apply to bucket {logBucketName} to
enable logging:");
    Console.WriteLine(newPolicy);

    PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
    {
        BucketName = logBucketName,
        Policy = newPolicy,
    };
    await client.PutBucketPolicyAsync(putRequest);
    Console.WriteLine("Policy applied.");
}

/// <summary>
/// This method enables logging for an Amazon S3 bucket. Logs will be
stored
/// in the bucket you selected for logging. Selected prefix
/// will be prepended to each log object.
/// </summary>
/// <param name="client">The initialized Amazon S3 client which will be
used
/// to configure and apply logging to the selected Amazon S3 bucket.</
param>
/// <param name="bucketName">The name of the Amazon S3 bucket for which
you
/// wish to enable logging.</param>
/// <param name="logBucketName">The name of the Amazon S3 bucket where
logging
/// information will be stored.</param>
/// <param name="logObjectKeyPrefix">The prefix to prepend to each
/// object key.</param>
/// <returns>Async task.</returns>

```

```

public static async Task EnableLoggingAsync(
    IAmazonS3 client,
    string bucketName,
    string logBucketName,
    string logObjectKeyPrefix)
{
    Console.WriteLine($"Enabling logging for bucket {bucketName}.");
    var loggingConfig = new S3BucketLoggingConfig
    {
        TargetBucketName = logBucketName,
        TargetPrefix = logObjectKeyPrefix,
    };

    var putBucketLoggingRequest = new PutBucketLoggingRequest
    {
        BucketName = bucketName,
        LoggingConfig = loggingConfig,
    };
    await client.PutBucketLoggingAsync(putBucketLoggingRequest);
    Console.WriteLine($"Logging enabled.");
}

/// <summary>
/// Loads configuration from settings files.
/// </summary>
public static void LoadConfig()
{
    _configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load settings from .json file.
        .AddJsonFile("settings.local.json", true) // Optionally, load
local settings.
        .Build();
}
}

```

- API-Einheiten finden Sie [PutBucketLogging](#) unter AWS SDK for .NET API-Referenz.

Java

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketLoggingStatus;
import software.amazon.awssdk.services.s3.model.LoggingEnabled;
import software.amazon.awssdk.services.s3.model.PartitionedPrefix;
import software.amazon.awssdk.services.s3.model.PutBucketLoggingRequest;
import software.amazon.awssdk.services.s3.model.TargetObjectKeyFormat;

// Class to set a bucket policy on a target S3 bucket and enable server access
logging on a source S3 bucket.
public class ServerAccessLogging {
    private static S3Client s3Client;

    public static void main(String[] args) {
        String sourceBucketName = "SOURCE-BUCKET";
        String targetBucketName = "TARGET-BUCKET";
        String sourceAccountId = "123456789012";
        String targetPrefix = "logs/";

        // Create S3 Client.
        s3Client = S3Client.builder().
            region(Region.US_EAST_2)
            .build();

        // Set a bucket policy on the target S3 bucket to enable server access
logging by granting the
        // logging.s3.amazonaws.com principal permission to use the PutObject
operation.
        ServerAccessLogging serverAccessLogging = new ServerAccessLogging();
        serverAccessLogging.setTargetBucketPolicy(sourceAccountId, sourceBucketName,
targetBucketName);

        // Enable server access logging on the source S3 bucket.
        serverAccessLogging.enableServerAccessLogging(sourceBucketName,
targetBucketName,
            targetPrefix);
    }

    // Function to set a bucket policy on the target S3 bucket to enable server
access logging by granting the
    // logging.s3.amazonaws.com principal permission to use the PutObject operation.
    public void setTargetBucketPolicy(String sourceAccountId, String
sourceBucketName, String targetBucketName) {
        String policy = "{\n" +
```

```

        "    \"Version\": \"2012-10-17\", \"n\" +
        "    \"Statement\": [\"n\" +
        "        {\"n\" +
        "            \"Sid\": \"S3ServerAccessLogsPolicy\", \"n\" +
        "            \"Effect\": \"Allow\", \"n\" +
        "            \"Principal\": {\"Service\": \"logging.s3.amazonaws.com
\"}, \"n\" +
        "            \"Action\": [\"n\" +
        "                \"s3:PutObject\" \"n\" +
        "            ], \"n\" +
        "            \"Resource\": \"arn:aws:s3::\" + targetBucketName + \"/*
\", \"n\" +
        "            \"Condition\": {\"n\" +
        "                \"ArnLike\": {\"n\" +
        "                    \"aws:SourceArn\": \"arn:aws:s3::\" +
sourceBucketName + \"\" \"n\" +
        "                }, \"n\" +
        "                \"StringEquals\": {\"n\" +
        "                    \"aws:SourceAccount\": \"\" + sourceAccountId +
\"\" \"n\" +
        "                } \"n\" +
        "            } \"n\" +
        "        ] \"n\" +
        "    } \"n\" +
        "};
        s3Client.putBucketPolicy(b -> b.bucket(targetBucketName).policy(policy));
    }

    // Function to enable server access logging on the source S3 bucket.
    public void enableServerAccessLogging(String sourceBucketName, String
targetBucketName,
        String targetPrefix) {
        TargetObjectKeyFormat targetObjectKeyFormat =
TargetObjectKeyFormat.builder()
.partitionedPrefix(PartitionedPrefix.builder().partitionDateSource("EventTime").build())
        .build();
        LoggingEnabled loggingEnabled = LoggingEnabled.builder()
            .targetBucket(targetBucketName)
            .targetPrefix(targetPrefix)
            .targetObjectKeyFormat(targetObjectKeyFormat)
            .build();
        BucketLoggingStatus bucketLoggingStatus = BucketLoggingStatus.builder()
            .loggingEnabled(loggingEnabled)

```

```
        .build();
    s3Client.putBucketLogging(PutBucketLoggingRequest.builder()
        .bucket(sourceBucketName)
        .bucketLoggingStatus(bucketLoggingStatus)
        .build());
}
}
```

Verwendung der AWS CLI

Wir empfehlen, dass Sie in jedem Bucket einen eigenen Logging-Bucket erstellen AWS-Region in dem Sie S3-Buckets haben. Anschließend können Sie Ihre Amazon-S3-Zugriffsprotokolle in diesem S3-Bucket bereitstellen lassen. Weitere Informationen und Beispiele finden Sie unter [put-bucket-logging](#) in der AWS CLI Referenz.

Wenn der Ziel-Bucket (Ziel-Bucket) die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum verwendet, können Sie weder Bucket noch Objekt ACLs festlegen. Sie können auch keine Zielzuschüsse (Zielzuschüsse) in Ihre [PutBucketLogging](#) Konfiguration aufnehmen. Sie müssen eine Bucket-Richtlinie verwenden, um dem Protokollierungsserviceprinzipal (`logging.s3.amazonaws.com`) Zugriff zu gewähren. Weitere Informationen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Example – Aktivieren von Zugriffsprotokollen für fünf Buckets in zwei Regionen

In diesem Beispiel werden die folgenden fünf Buckets verwendet:

- 1-amzn-s3-demo-bucket1-us-east-1
- 2-amzn-s3-demo-bucket1-us-east-1
- 3-amzn-s3-demo-bucket1-us-east-1
- 1-amzn-s3-demo-bucket1-us-west-2
- 2-amzn-s3-demo-bucket1-us-west-2

Note

Der letzte Schritt im folgenden Verfahren enthält Bash-Beispielskripts, mit denen Sie Ihre Protokollierungs-Buckets erstellen und die Serverzugriffsprotokollierung für diese Buckets

aktivieren können. Um diese Skripts verwenden zu können, müssen Sie die Dateien `policy.json` und `logging.json` erstellen wie im folgenden Verfahren beschrieben.

1. Erstellen Sie in den Regionen USA West (Oregon) und USA Ost (Nord-Virginia) zwei Ziel-Buckets für die Protokollierung mit den folgenden Namen:
 - `amzn-s3-demo-bucket1-logs-us-east-1`
 - `amzn-s3-demo-bucket1-logs-us-west-2`
2. Später im Verfahren aktivieren Sie die Serverzugriffsprotokollierung wie folgt:
 - `1-amzn-s3-demo-bucket1-us-east-1` protokolliert zum S3-Bucket `amzn-s3-demo-bucket1-logs-us-east-1` mit dem Präfix `1-amzn-s3-demo-bucket1-us-east-1`
 - `2-amzn-s3-demo-bucket1-us-east-1` protokolliert zum S3-Bucket `amzn-s3-demo-bucket1-logs-us-east-1` mit dem Präfix `2-amzn-s3-demo-bucket1-us-east-1`
 - `3-amzn-s3-demo-bucket1-us-east-1` protokolliert zum S3-Bucket `amzn-s3-demo-bucket1-logs-us-east-1` mit dem Präfix `3-amzn-s3-demo-bucket1-us-east-1`
 - `1-amzn-s3-demo-bucket1-us-west-2` protokolliert zum S3-Bucket `amzn-s3-demo-bucket1-logs-us-west-2` mit dem Präfix `1-amzn-s3-demo-bucket1-us-west-2`
 - `2-amzn-s3-demo-bucket1-us-west-2` protokolliert zum S3-Bucket `amzn-s3-demo-bucket1-logs-us-west-2` mit dem Präfix `2-amzn-s3-demo-bucket1-us-west-2`
3. Erteilen Sie für jeden Ziel-Logging-Bucket mithilfe eines Buckets ACL oder einer Bucket-Richtlinie Berechtigungen für die Übermittlung von Serverzugriffsprotokollen:
 - Aktualisieren Sie die Bucket-Richtlinie (Empfohlen) – Verwenden Sie den folgenden `put-bucket-policy`-Befehl, um dem Prinzipal des Protokollierungsservices Berechtigungen zu erteilen. Ersetzen `amzn-s3-demo-destination-bucket-Protokolle` mit dem Namen Ihres Ziel-Buckets.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-destination-bucket-logs --  
policy file://policy.json
```

`Policy.json` ist ein JSON Dokument im aktuellen Ordner, das die folgende Bucket-Richtlinie enthält. Wenn Sie diese Bucket-Richtlinie verwenden möchten, ersetzen Sie `user input placeholders` durch Ihre eigenen Informationen. In der folgenden Richtlinie `amzn-s3-demo-destination-bucket-Protokolle` ist der Ziel-Bucket, in den

Serverzugriffsprotokolle übermittelt werden, und *amzn-s3-demo-source-bucket* ist der Quell-Bucket. *SOURCE-ACCOUNT-ID* ist der AWS-Konto dem der Quell-Bucket gehört.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3ServerAccessLogsPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "logging.s3.amazonaws.com"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket-logs/*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
        },
        "StringEquals": {
          "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
        }
      }
    }
  ]
}
```

- Bucket aktualisieren ACL — Verwenden Sie den folgenden `put-bucket-acl` Befehl, um der S3-Protokollzustellungsgruppe Berechtigungen zu erteilen. Ersetzen *amzn-s3-demo-destination-bucket-loggt* mit dem Namen Ihres Ziel-Buckets (Ziel-Buckets).

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-destination-bucket-logs --
grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp
URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

4. Erstellen Sie dann eine `logging.json`-Datei, die Ihre Protokollierungskonfiguration enthält (basierend auf einem der drei folgenden Beispiele). Nach der Erstellung der

logging.json-Datei können Sie mit dem folgenden put-bucket-logging-Befehl die Protokollierungskonfiguration anwenden. Ersetzen *amzn-s3-demo-destination-bucket-Logs* mit dem Namen Ihres Ziel-Buckets (Ziel-Buckets).

```
aws s3api put-bucket-logging --bucket amzn-s3-demo-destination-bucket-logs --  
bucket-logging-status file://logging.json
```

Note

Anstatt mit diesem diesen put-bucket-logging-Befehl die Protokollierungskonfiguration auf jeden einzelnen Ziel-Bucket anzuwenden, können Sie eines der im nächsten Schritt bereitgestellten Bash-Skripts verwenden. Um diese Skripts verwenden zu können, müssen Sie die Dateien policy.json und logging.json erstellen wie in diesem Verfahren beschrieben.

Die logging.json Datei ist ein JSON Dokument im aktuellen Ordner, das Ihre Logging-Konfiguration enthält. Wenn ein Ziel-Bucket die Einstellung „Von Bucket-Besitzer erzwungen“ für Object Ownership verwendet, kann Ihre Protokollierungskonfiguration keine Zielgewährungen enthalten. Weitere Informationen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Example – **logging.json** ohne Zielgewährungen

Die folgende logging.json-Beispieldatei enthält keine Zielgewährungen. Daher können Sie diese Konfiguration auf einen Ziel-Bucket anwenden, der die Einstellung „Von Bucket-Besitzer erzwungen“ für Object Ownership verwendet.

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "amzn-s3-demo-destination-bucket-logs",  
    "TargetPrefix": "amzn-s3-demo-destination-bucket/"  
  }  
}
```

Example – **logging.json** mit Zielgewährungen

Die folgende `logging.json`-Beispieldatei enthält Zielgewährungen.

Wenn der Ziel-Bucket die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum verwendet, können Sie Zielzuschüsse (Zielzuschüsse) nicht in Ihre [PutBucketLogging](#)-Konfiguration. Weitere Informationen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

```
{
  "LoggingEnabled": {
    "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
    "TargetPrefix": "amzn-s3-demo-destination-bucket/",
    "TargetGrants": [
      {
        "Grantee": {
          "Type": "AmazonCustomerByEmail",
          "EmailAddress": "user@example.com"
        },
        "Permission": "FULL_CONTROL"
      }
    ]
  }
}
```

Example – **logging.json** mit auf S3-Ereigniszeit festgelegtem Protokollobjekt-Schlüsselformat

In der folgenden `logging.json`-Datei wird das Protokollobjekt-Schlüsselformat in S3-Ereigniszeit geändert. Weitere Informationen zum Einrichten des Protokollobjekt-Schlüsselformats finden Sie unter [the section called “Wie aktiviere ich die Protokollzustellung?”](#).

```
{
  "LoggingEnabled": {
    "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
    "TargetPrefix": "amzn-s3-demo-destination-bucket/",
    "TargetObjectKeyFormat": {
      "PartitionedPrefix": {
        "PartitionDateSource": "EventTime"
      }
    }
  }
}
```

```

    }
  }
}

```

5. Sie können eines der folgenden Bash-Skripts verwenden, um allen Buckets in Ihrem Konto Zugriffsprotokollierung hinzuzufügen. Ersetzen *amzn-s3-demo-destination-bucket-logs* mit dem Namen Ihres Ziel-Buckets (Ziel-Buckets) und *us-west-2* ersetzen Sie es durch den Namen der Region, in der sich Ihre Buckets befinden.

Note

Dieses Skript funktioniert nur, wenn sich alle Buckets in derselben Region befinden. Sind die Buckets auf mehrere Regionen verteilt, müssen Sie das Skript anpassen.

Example - Gewähren Sie Zugriff mit Bucket-Richtlinien und fügen Sie die Protokollierung für die Buckets in Ihrem Konto hinzu

```

loggingBucket='amzn-s3-demo-destination-bucket-logs'
region='us-west-2'

# Create the logging bucket.
aws s3 mb s3://$loggingBucket --region $region

aws s3api put-bucket-policy --bucket $loggingBucket --policy file://policy.json

# List the buckets in this account.
buckets="$(aws s3 ls | awk '{print $3}')"

# Put a bucket logging configuration on each bucket.
for bucket in $buckets
do
    # This if statement excludes the logging bucket.
    if [ "$bucket" != "$loggingBucket" ] ; then
        continue;
    fi
    printf '{
        "LoggingEnabled": {
            "TargetBucket": "%s",

```

```

        "TargetPrefix": "%s/"
    }
}' "$loggingBucket" "$bucket" > logging.json
aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
echo "$bucket done"
done

rm logging.json

echo "Complete"

```

Example — Gewähren Sie Zugriff mit einem Bucket ACLs und fügen Sie die Protokollierung für die Buckets in Ihrem Konto hinzu

```

loggingBucket='amzn-s3-demo-destination-bucket-logs'
region='us-west-2'

# Create the logging bucket.
aws s3 mb s3://$loggingBucket --region $region

aws s3api put-bucket-acl --bucket $loggingBucket --grant-write URI=http://
acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://
acs.amazonaws.com/groups/s3/LogDelivery

# List the buckets in this account.
buckets="$(aws s3 ls | awk '{print $3}')"

# Put a bucket logging configuration on each bucket.
for bucket in $buckets
do
    # This if statement excludes the logging bucket.
    if [ "$bucket" != "$loggingBucket" ] ; then
        continue;
    fi
    printf '{
        "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
    }' "$loggingBucket" "$bucket" > logging.json

```

```
aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://  
logging.json  
echo "$bucket done"  
done  
  
rm logging.json  
  
echo "Complete"
```

Überprüfen der Einrichtung Ihrer Serverzugriffsprotokolle

Führen Sie nach dem Aktivieren der Serverzugriffsprotokollierung die folgenden Schritte aus:

- Greifen Sie auf den Ziel-Bucket zu und überprüfen Sie, ob die Protokolldateien bereitgestellt werden. Nachdem die Zugriffsprotokolle eingerichtet wurden, fängt Amazon S3 sofort an, Anfragen zu erfassen und zu protokollieren. Es kann jedoch einige Stunden dauern, bis die Protokolle an den Ziel-Bucket übermittelt werden. Weitere Informationen erhalten Sie unter [the section called “Statusänderungen in der Bucket-Protokollierung werden mit der Zeit wirksam”](#) und [the section called “Best-Effort-Serverprotokollbereitstellung”](#).

Sie können die Protokollzustellung auch automatisch überprüfen, indem Sie Amazon S3 S3-Anforderungsmetriken verwenden und CloudWatch Amazon-Alarme für diese Metriken einrichten. Weitere Informationen finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

- Vergewissern Sie sich, dass Sie den Inhalt der Protokolldateien öffnen und lesen können.

Informationen zur Fehlerbehebung bei der Serverzugriffsprotokollierung finden Sie unter [Behebung von Fehlern bei der Server-Zugriffsprotokollierung](#).

Amazon-S3-Server-Zugriffsprotokollformat

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an einen Amazon-S3-Bucket gestellt wurden. Sie können Serverzugriffsprotokolle für folgende Zwecke verwenden:

- Durchführung von Sicherheits- und Zugriffsprüfungen
- Kennenlernen Ihres Kundenstamms
- Nachvollziehen Ihrer Amazon-S3-Rechnung

Dieser Abschnitt beschreibt das Format und andere Details zu Amazon-S3-Serverzugriffprotokolldateien.

Die Server-Zugriffsprotokolldateien bestehen aus einer Reihe von durch Zeilenschaltungen voneinander getrennten Protokolldatensätzen. Jeder Protokolldatensatz stellt eine Anforderung dar und besteht aus durch Leerzeichen voneinander getrennter Felder.

Nachfolgend wird ein Beispielprotokoll mit sechs Protokolldatensätzen gezeigt.

```

79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
  amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 - 113 - 7 -
 "-" "S3Console/0.4" - s9lzHYrFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
XV/VLi31234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-demo-bucket1.s3.us-
west-1.amazonaws.com TLSV1.2 arn:aws:s3:us-west-1:123456789012:accesspoint/example-AP
Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
  amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
REST.GET.LOGGING_STATUS - "GET /amzn-s3-demo-bucket1?logging HTTP/1.1" 200 -
 242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mX0cqPGzQ0I5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBU0ZJG96N1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
  amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
REST.GET.BUCKETPOLICY - "GET /amzn-s3-demo-bucket1?policy HTTP/1.1" 404
NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV4 ECDHE-RSA-AES128-GCM-SHA256
AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
  amzn-s3-demo-bucket1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 -
 113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuU1PJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpbybfEseEME/u7ME1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
  amzn-s3-demo-bucket1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
  DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /amzn-s3-demo-bucket1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -

```

```
10S62Zv81kBW7BB6SX4XJ48o6kpc16LPwEoizZQ0xJd5qDSCTLX0TgS37kYUBKQW3+bPdrG1234= SigV4
ECDHE-RSA-AES128-SHA AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com
TLV1.2 - Yes
```

Note

Jedes Feld kann auf - gesetzt werden, um darauf hinzuweisen, dass die Daten unbekannt oder nicht verfügbar waren, oder dass das Feld für diese Anforderung nicht relevant war.

Themen

- [Protokolldatensatzfelder](#)
- [Zusätzliche Protokollierung für Kopiervorgänge](#)
- [Benutzerdefinierte Zugriffsprotokollinformationen](#)
- [Aspekte zur Programmierung des erweiterbaren Serverzugriff-Protokollformats](#)

Protokolldatensatzfelder

In der folgenden Liste werden die wichtigsten Protokolldatensatzfelder beschrieben.

Bucket-Eigentümer

Die kanonische Benutzer-ID des Eigentümers des Quell-Buckets. Die kanonische Benutzer-ID ist eine andere Form von AWS-Konto ID. Weitere Informationen zur kanonischen Benutzer-ID finden Sie unter [AWS-Konto Bezeichner](#) in der Allgemeinen AWS-Referenz. Informationen darüber, wie Sie die kanonische Benutzer-ID für Ihr Konto finden, finden Sie unter [Finden der kanonischen Benutzer-ID für Ihr AWS-Konto](#).

Beispielintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Der Name des Buckets, für den die Anforderung verarbeitet wurde. Wenn das System eine fehlerhaft aufgebaute Anforderung erhält und den Bucket nicht bestimmen kann, erscheint die Anforderung nicht in einem Server-Zugriffsprotokoll.

Beispielintrag

Zeit

Die Uhrzeit, zu der die Anfrage eingegangen ist. Diese Daten und Uhrzeiten sind in koordinierter Weltzeit () angegeben. UTC Das Format unter Verwendung der `strftime()`-Terminologie, nämlich: [%d/%b/%Y:%H:%M:%S %z]

Beispielintrag

```
[06/Feb/2019:00:00:38 +0000]
```

Remote-IP

Die offensichtliche IP-Adresse des Anforderers. Auf dem Weg vorhandene Proxy-Server und Firewalls könnten die tatsächliche IP-Adresse des Computers verbergen, der die Anforderung gestellt hat.

Beispielintrag

```
192.0.2.3
```

Auftraggeber

Die kanonische Benutzer-ID des Auftraggebers, oder - für nicht authentifizierte Anforderungen. Wenn der Anforderer ein IAM Benutzer war, gibt dieses Feld den IAM Benutzernamen des Anfragenden zusammen mit dem Root-Benutzer des AWS-Kontos zu dem der IAM Benutzer gehört. Diese ID ist dieselbe, die für den Zugriff zu Kontrollzwecken verwendet wird.

Beispielintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Wenn der Anforderer eine angenommene Rolle verwendet, gibt dieses Feld die angenommene IAM Rolle zurück.

Beispielintrag

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

Anforderungs-ID

Eine von Amazon S3 generierte Zeichenfolge, die jede Anforderung eindeutig identifiziert.

Beispielintrag

```
3E57427F33A59F07
```

Operation

Die hier aufgeführte Operation ist deklariert als SOAP.*operation*, REST.*HTTP_method.resource_type*, WEBSITE.*HTTP_method.resource_type* oder BATCH.DELETE.OBJECT, oder S3.action.resource_type für [Lebenszyklus und Protokollieren](#).

Beispielintrag

```
REST.PUT.OBJECT
```

Schlüssel

Der Schlüsselteil (Objektname) der Anfrage.

Beispielintrag

```
/photos/2019/08/puppy.jpg
```

Request-URI

Der Request-URI Teil der HTTP Anforderungsnachricht.

Beispielintrag

```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

HTTPStatus

Der numerische HTTP Statuscode der Antwort.

Beispielintrag

```
200
```

Fehlercode

Der Amazon S3 [Fehlercode](#) oder -, wenn kein Fehler aufgetreten ist.

Beispieleintrag

```
NoSuchBucket
```

Gesendete Bytes

Die Anzahl der gesendeten Antwortbytes ohne HTTP Protokoll-Overhead oder - falls Null.

Beispieleintrag

```
2662992
```

Objektgröße

Die Gesamtgröße des betreffenden Objekts.

Beispieleintrag

```
3462992
```

Gesamtzeit

Die Anzahl der Millisekunden (ms), die die Anforderung aus Perspektive des Servers unterwegs war. Dieser Wert wird ab der Zeit gemessen, zu der Ihre Anforderung empfangen wurde, bis zu der Zeit, zu der das letzte Byte der Antwort gesendet wurde. Messungen aus der Perspektive des Clients dauern möglicherweise länger aufgrund der Netzwerklatenz.

Beispieleintrag

```
70
```

Umschlagzeit

Die Anzahl der Millisekunden, die Amazon S3 gebraucht hat, Ihre Anfrage zu verarbeiten. Dieser Wert wird ab der Zeit gemessen, zu der das letzte Byte Ihrer Anforderung empfangen wurde, bis zu der Zeit, zu der das erste Byte der Antwort gesendet wurde.

Beispieleintrag

10

Referer

Der Wert des HTTP `Referer` Headers, falls vorhanden. HTTPBenutzeragenten (z. B. Browser) setzen diesen Header normalerweise auf den Header URL der verlinkenden oder einbettenden Seite, wenn sie eine Anfrage stellen.

Beispielintrag

```
"http://www.example.com/webservices"
```

User-Agent

Der Wert des Headers. HTTP `User-Agent`

Beispielintrag

```
"curl/7.15.1"
```

Versions-ID

Die Versions-ID der Anforderung oder -, wenn die Operation keinen `versionId`-Parameter entgegennimmt.

Beispielintrag

```
3HL4kqtJvjVBH40NrjfkD
```

Host-ID

Die erweiterte Anforderungs-ID für `x-amz-id-2` oder Amazon S3

Beispielintrag

```
s9lzhYrFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signatur-Version

Die Signaturversion, `SigV2` oder `SigV4`, die für die Authentifizierung der Anforderung verwendet wurde, bzw. ein - für nicht authentifizierte Anforderungen.

Beispielintrag

```
SigV2
```

Cipher Suite

Die Secure Sockets Layer (SSL) -Verschlüsselung, die für eine HTTPS Anfrage oder ein - für ausgehandelt wurde. HTTP

Beispielintrag

```
ECDHE-RSA-AES128-GCM-SHA256
```

Authentifizierungstyp

Die Art der verwendeten Anforderungsauthentifizierung: AuthHeader für Authentifizierungs-Header, QueryString für Abfragezeichenfolgen (vorsigniertURL) oder a - für nicht authentifizierte Anfragen.

Beispielintrag

```
AuthHeader
```

Host Header

Der für die Verbindung mit Amazon S3 verwendete Endpunkt.

Beispielintrag

```
s3.us-west-2.amazonaws.com
```

Einige frühere Regionen unterstützen Legacy-Endpunkte. Möglicherweise sehen Sie diese Endpunkte in Ihren Serverzugriffsprotokollen oder AWS CloudTrail Protokolle. Weitere Informationen finden Sie unter [Legacy-Endpunkte](#). Eine vollständige Liste der Amazon S3 S3-Regionen und -Endpunkte finden Sie unter [Amazon S3 S3-Endpunkte und Kontingente](#) in der Allgemeine Amazon Web Services-Referenz.

TLSVersion

Die vom Client ausgehandelte Version von Transport Layer Security (TLS). Der Wert ist einer der folgenden: TLSv1.1., TLSv1.2TLSv1.3, oder - falls TLS er nicht verwendet wurde.

Beispieleintrag

```
TLSv1.2
```

Zugriffspunkt ARN

Der Amazon-Ressourcenname (ARN) des Zugriffspunkts der Anfrage. Wenn der Access Point falsch ARN formatiert ist oder nicht verwendet wird, enthält das Feld eine-. Weitere Hinweise zu Zugangspunkten finden Sie unter [Verwenden von Zugriffspunkten](#). Weitere Informationen zu ARNs finden Sie unter [Amazon Resource Name \(ARN\)](#) in der AWS Referenzhandbuch.

Beispieleintrag

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Eine Zeichenfolge, die angibt, ob für die Anforderung eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich war. Wenn ACL für die Anfrage eine Autorisierung erforderlich war, lautet die Zeichenfolge Yes. Wenn keine erforderlichen ACLs waren, lautet die Zeichenfolge -. Weitere Informationen zu finden ACLs Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#). Weitere Informationen zur Verwendung des aclRequired Felds zum Deaktivieren ACLs finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Beispieleintrag

```
Yes
```

Zusätzliche Protokollierung für Kopiervorgänge

Eine Kopieroperation umfasst ein GET und ein PUT. Aus diesem Grund zeichnen wir für eine Kopieroperation zwei Datensätze auf. Der vorherige Abschnitt beschreibt die Felder für den PUT-Teil der Operation. Die folgende Liste beschreibt die Felder in dem Datensatz, die sich auf den GET-Teil der Kopieroperation beziehen.

Bucket-Eigentümer

Die kanonische Benutzer-ID des Buckets, der das kopierte Objekt speichert. Die kanonische Benutzer-ID ist eine andere Form von AWS-Konto ID. Weitere Informationen zur kanonischen Benutzer-ID finden Sie unter [AWS-Konto Bezeichner](#) in der Allgemeine AWS-Referenz.

Informationen darüber, wie Sie die kanonische Benutzer-ID für Ihr Konto finden, finden Sie unter [Finden der kanonischen Benutzer-ID für Ihr AWS-Konto](#).

Beispieleintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Der Name des Buckets, der das kopierte Objekt speichert.

Beispieleintrag

Zeit

Die Uhrzeit, zu der die Anfrage eingegangen ist. Diese Daten und Uhrzeiten sind in koordinierter Weltzeit (UTC) angegeben. Das Format unter Verwendung der `strftime()`-Terminologie, nämlich: `[%d/%B/%Y:%H:%M:%S %z]`

Beispieleintrag

```
[06/Feb/2019:00:00:38 +0000]
```

Remote-IP

Die offensichtliche IP-Adresse des Anforderers. Auf dem Weg vorhandene Proxy-Server und Firewalls könnten die tatsächliche IP-Adresse des Computers verbergen, der die Anforderung gestellt hat.

Beispieleintrag

```
192.0.2.3
```

Auftraggeber

Die kanonische Benutzer-ID des Auftraggebers, oder - für nicht authentifizierte Anforderungen. Wenn der Anforderer ein IAM Benutzer war, gibt dieses Feld den IAM Benutzernamen des Anfragenden zusammen mit dem Root-Benutzer des AWS-Kontos zu dem der IAM Benutzer gehört. Diese ID ist dieselbe, die für den Zugriff zu Kontrollzwecken verwendet wird.

Beispieleintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Wenn der Anforderer eine angenommene Rolle verwendet, gibt dieses Feld die angenommene IAM Rolle zurück.

Beispieleintrag

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

Anforderungs-ID

Eine von Amazon S3 generierte Zeichenfolge, die jede Anforderung eindeutig identifiziert.

Beispieleintrag

```
3E57427F33A59F07
```

Operation

Die hier aufgeführte Operation ist deklariert als SOAP.*operation*, REST.*HTTP_method.resource_type*, WEBSITE.*HTTP_method.resource_type* oder BATCH.DELETE.OBJECT.

Beispieleintrag

```
REST.COPY.OBJECT_GET
```

Schlüssel

Der Schlüssel (Objektname) des kopierten Objekts oder -, wenn die Operation keinen Schlüsselparameter entgegennimmt.

Beispieleintrag

```
/photos/2019/08/puppy.jpg
```

Request-URI

Der Request-URI Teil der HTTP Anforderungsnachricht.

Beispieleintrag


```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar"
```

HTTPStatus

Der numerische HTTP Statuscode des GET Teils des Kopiervorgangs.

Beispielintrag

```
200
```

Fehlercode

Der [Fehlercode](#) von Amazon S3 des GET-Teils des Kopiervorgangs oder -, wenn kein Fehler aufgetreten ist.

Beispielintrag

```
NoSuchBucket
```

Gesendete Bytes

Die Anzahl der gesendeten Antwortbytes, ohne den HTTP Protokoll-Overhead, oder - wenn Null.

Beispielintrag

```
2662992
```

Objektgröße

Die Gesamtgröße des betreffenden Objekts.

Beispielintrag

```
3462992
```

Gesamtzeit

Die Anzahl der Millisekunden (ms), die die Anforderung aus Perspektive des Servers unterwegs war. Dieser Wert wird ab der Zeit gemessen, zu der Ihre Anforderung empfangen wurde, bis zu der Zeit, zu der das letzte Byte der Antwort gesendet wurde. Messungen aus der Perspektive des Clients dauern möglicherweise länger aufgrund der Netzwerklatenz.

Beispielintrag

```
70
```

Umschlagzeit

Die Anzahl der Millisekunden, die Amazon S3 gebraucht hat, Ihre Anfrage zu verarbeiten. Dieser Wert wird ab der Zeit gemessen, zu der das letzte Byte Ihrer Anforderung empfangen wurde, bis zu der Zeit, zu der das erste Byte der Antwort gesendet wurde.

Beispielintrag

```
10
```

Referer

Der Wert des HTTP `Referer` Headers, falls vorhanden. HTTPBenutzeragenten (z. B. Browser) setzen diesen Header normalerweise auf den Header URL der verlinkenden oder einbettenden Seite, wenn sie eine Anfrage stellen.

Beispielintrag

```
"http://www.example.com/webservices"
```

User-Agent

Der Wert des Headers. HTTP `User-Agent`

Beispielintrag

```
"curl/7.15.1"
```

Versions-ID

Die Version-ID des kopierten Objekts oder -, wenn der `x-amz-copy-source`-Header keinen `versionId`-Parameter als Teil der Kopierquelle angegeben hat.

Beispielintrag

```
3HL4kqtJvjVBH40N1jfkD
```

Host-ID

Die erweiterte Anforderungs-ID für `x-amz-id-2` oder Amazon S3

Beispielintrag

```
s91zHYrFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signatur-Version

Die Signaturversion, `SigV2` oder `SigV4`, die für die Authentifizierung der Anforderung verwendet wurde, bzw. ein `-` für nicht authentifizierte Anforderungen.

Beispielintrag

```
SigV4
```

Cipher Suite

Die Secure Sockets Layer (SSL) -Chiffre, die für eine HTTPS Anfrage ausgehandelt wurde, oder eine `-` für HTTP

Beispielintrag

```
ECDHE-RSA-AES128-GCM-SHA256
```

Authentifizierungstyp

Die Art der verwendeten Anforderungsauthentifizierung: `AuthHeader` für Authentifizierungs-Header, `QueryString` für Abfragezeichenfolgen (vorsigniertURLs) oder `a` - für nicht authentifizierte Anfragen.

Beispielintrag

```
AuthHeader
```

Host-Header

Der für die Verbindung mit Amazon S3 verwendete Endpunkt.

Beispielintrag

```
s3.us-west-2.amazonaws.com
```

Einige frühere Regionen unterstützen Legacy-Endpunkte. Möglicherweise sehen Sie diese Endpunkte in Ihren Serverzugriffsprotokollen oder AWS CloudTrail Protokolle. Weitere Informationen finden Sie unter [Legacy-Endpunkte](#). Eine vollständige Liste der Amazon S3 S3-Regionen und -Endpunkte finden Sie unter [Amazon S3 S3-Endpunkte und Kontingente](#) in der Allgemeine Amazon Web Services-Referenz.

TLSVersion

Die vom Client ausgehandelte Version von Transport Layer Security (TLS). Der Wert ist einer der folgenden: TLSv1.1, TLSv1.2, TLSv1.3, oder - falls TLS er nicht verwendet wurde.

Beispieleintrag

```
TLSv1.2
```

Zugriffspunkt ARN

Der Amazon-Ressourcenname (ARN) des Zugriffspunkts der Anfrage. Wenn der Access Point falsch ARN formatiert ist oder nicht verwendet wird, enthält das Feld eine -. Weitere Hinweise zu Zugangspunkten finden Sie unter [Verwenden von Zugriffspunkten](#). Weitere Informationen zu ARNs finden Sie unter [Amazon Resource Name \(ARN\)](#) in der AWS Referenzhandbuch.

Beispieleintrag

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Eine Zeichenfolge, die angibt, ob für die Anforderung eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich war. Wenn ACL für die Anfrage eine Autorisierung erforderlich war, lautet die Zeichenfolge Yes. Wenn keine erforderlich ACLs waren, lautet die Zeichenfolge -. Weitere Informationen zu finden ACLs Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#). Weitere Informationen zur Verwendung des aclRequired Felds zum Deaktivieren ACLs finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Beispieleintrag

```
Yes
```

Benutzerdefinierte Zugriffsprotokollinformationen

Sie können benutzerdefinierte Informationen angeben, die im Zugriffsprotokoll Datensatz für eine Anforderung gespeichert werden. Fügen Sie dazu dem URL für die Anforderung einen benutzerdefinierten Abfragezeichenfolge-Parameter hinzu. Amazon S3 ignoriert Abfrage-Zeichenfolgenparameter, die mit x- beginnen, aber nimmt diese in den Zugriffsprotokoll-Datensatz für die Anforderung auf, als Teil des Request-URI-Felds des Protokoll Datensatzes.

Beispielsweise verhält sich die Anfrage GET für "s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-user=johndoe" genauso wie die Anfrage für "s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg", abgesehen davon, dass die Zeichenfolge "x-user=johndoe" in das Feld Request-URI des entsprechenden Protokoll Datensatzes eingefügt wird. Diese Funktionalität ist nur in der REST Benutzeroberfläche verfügbar.

Aspekte zur Programmierung des erweiterbaren Serverzugriff-Protokollformats

Gelegentlich erweitern wir das Zugriffsprotokoll-Datensatzformat, indem wir am Ende jeder Zeile neue Felder hinzufügen. Daher sollten Sie sicherstellen, dass jeder Code, der Server-Zugriffsprotokolle analysiert, so geschrieben ist, dass er angefügte Felder verarbeiten kann, die er möglicherweise nicht versteht.

Löschen von Amazon-S3-Protokolldateien

Ein Amazon-S3-Bucket mit aktivierter Server-Zugriffsprotokollierung kann im Laufe der Zeit viele Server-Protokollobjekte ansammeln. Möglicherweise benötigt Ihre Anwendung diese Zugriffsprotokolle für einen bestimmten Zeitraum nachdem sie erstellt wurden und danach möchten Sie sie ggf. löschen. Sie können die Amazon-S3-Lebenszykluskonfiguration zum Erstellen von Regeln verwenden, sodass Amazon S3 diese Objekte am Ende ihres Lebenszyklus automatisch zum Löschen bereitstellt.

Sie können mithilfe eines gemeinsamen Präfixes für eine Teilmenge der Objekte in Ihrem S3-Bucket eine Lebenszykluskonfiguration definieren. Wenn Sie in Ihrer Konfiguration für die Server-Zugriffsprotokollierung ein Präfix angegeben haben, können Sie eine Lebenszyklus-Konfigurationsregel festlegen, um Protokollobjekte mit diesem Präfix zu löschen.

Angenommen, Ihre Protokollobjekte verwenden das Präfix logs/. Sie können dann über eine Lebenszyklus-Konfigurationsregel alle Objekte mit dem Präfix logs/ im Bucket nach einem bestimmten Zeitraum löschen.

Weitere Informationen zur Lebenszyklus-Konfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Allgemeine Informationen zu Server-Zugriffsprotokollen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

Verwenden von Amazon-S3-Serverzugriffsprotokollen zur Identifizierung von Anforderungen

Sie können mittels Amazon-S3-Serverzugriffsprotokollen Amazon-S3-Anforderungen identifizieren.

Note

- Um Amazon S3 S3-Anfragen zu identifizieren, empfehlen wir die Verwendung von AWS CloudTrail Datenereignisse anstelle von Amazon S3 S3-Serverzugriffsprotokollen. CloudTrail Datenereignisse sind einfacher einzurichten und enthalten mehr Informationen. Weitere Informationen finden Sie unter [Identifizieren von Amazon S3 S3-Anfragen mit CloudTrail](#).
- Je nachdem, wie viele Zugriffsanfragen Sie erhalten, erfordert die Analyse Ihrer Protokolle möglicherweise mehr Ressourcen oder Zeit als die Verwendung von CloudTrail Datenereignissen.

Themen

- [Abfragen von Zugriffsprotokollen für Anforderungen über Amazon Athena](#)
- [Identifizieren von Signature-Version-2-Anforderungen mittels Amazon-S3-Zugriffsprotokollen](#)
- [Identifizieren von Objektzugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen](#)

Abfragen von Zugriffsprotokollen für Anforderungen über Amazon Athena

Sie können Amazon-S3-Anforderungen mit Amazon-S3-Zugriffsprotokollen mithilfe von Amazon Athena identifizieren.

Amazon S3 speichert Server-Zugriffsprotokolle als Objekte in einem S3-Bucket. Es ist oft einfacher, ein Tool zu verwenden, mit dem die Protokolle in Amazon S3 analysiert werden können. Athena unterstützt die Analyse von S3-Objekten und kann zur Abfrage von Amazon-S3-Zugriffsprotokollen verwendet werden.

Example

Das folgende Beispiel zeigt, wie Sie Amazon-S3-Server-Zugriffsprotokolle in Amazon Athena abfragen können. Ersetzen Sie die *user input placeholders* in den folgenden Beispielen durch eigene Daten.

Note

Um einen Amazon S3 S3-Standort in einer Athena-Abfrage anzugeben, müssen Sie einen S3 URI für den Bucket angeben, in den Ihre Logs geliefert werden. Dies URI muss den Bucket-Namen und das Präfix im folgenden Format enthalten: `s3://amzn-s3-demo-bucket1-logs/prefix/`

1. Öffnen Sie die Athena-Konsole unter <https://console.aws.amazon.com/athena/>.
2. Führen Sie im Abfrage-Editor einen Befehl wie den folgenden aus. Ersetzen Sie `s3_access_logs_db` durch den Namen, den Sie Ihrer Datenbank geben möchten.

```
CREATE DATABASE s3_access_logs_db
```

Note

Es hat sich bewährt, die Datenbank im selben Format zu erstellen AWS-Region wie Ihr S3-Bucket.

3. Führen Sie im Abfrage-Editor einen Befehl wie den folgenden aus, um in der in Schritt 2 erstellten Datenbank ein Tabellenschema zu erstellen. Ersetzen Sie `s3_access_logs_db.mybucket_logs` durch den Namen, den Sie Ihrer Tabelle geben möchten. Die Datentypwerte `STRING` und `BIGINT` sind die Zugriffseigenschaften. Sie können diese Eigenschaften in Athena abfragen. Geben Sie für `LOCATION` wie oben erwähnt den Pfad von S3-Bucket und Präfix ein.

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs` (  
  `bucketowner` STRING,  
  `bucket_name` STRING,  
  `requestdatetime` STRING,  
  `remoteip` STRING,  
  `requester` STRING,
```

```

`requestid` STRING,
`operation` STRING,
`key` STRING,
`request_uri` STRING,
`httpstatus` STRING,
`errorcode` STRING,
`bytessent` BIGINT,
`objectsize` BIGINT,
`totaltime` STRING,
`turnaroundtime` STRING,
`referrer` STRING,
`useragent` STRING,
`versionid` STRING,
`hostid` STRING,
`sigv` STRING,
`ciphersuite` STRING,
`authtype` STRING,
`endpoint` STRING,
`tlsversion` STRING,
`accesspointarn` STRING,
`aclrequired` STRING)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([ ]*) ([ ]*) \\[(.?)\\] ([ ]*) ([ ]*) ([ ]*) ([ ]*)
([ ]*) (\\\"[^\"]*\\\"|-) (-|[0-9]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*)
(\\\"[^\"]*\\\"|-) ([ ]*)(?: ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*)
([ ]*))?.*$')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://amzn-s3-demo-bucket1-logs/prefix/'

```

4. Wählen Sie im Navigationsbereich unter Database (Datenbank) die Datenbank aus.
5. Wählen Sie unter Tables (Tabellen) neben dem Namen der Tabelle Preview table (Tabellenvorschau) aus.

Im Fensterbereich Results (Ergebnisse) sollten Daten aus den Server-Zugriffsprotokollen angezeigt werden, also `bucketowner`, `bucket`, `requestdatetime` usw. Dies bedeutet, dass die Athena-Tabelle erfolgreich erstellt wurde. Sie können jetzt die Amazon-S3-Server-Zugriffsprotokolle abfragen.

Example — Zeigt an, wer ein Objekt wann gelöscht hat (Zeitstempel, IP-Adresse und IAM Benutzer)

```
SELECT requestdatetime, remoteip, requester, key
FROM s3_access_logs_db.mybucket_logs
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

Example — Zeigt alle Operationen an, die von einem IAM Benutzer ausgeführt wurden

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example – Anzeigen aller Vorgänge, die in einem bestimmten Zeitraum für ein Objekt ausgeführt wurden

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE Key='prefix/images/picture.jpg'
      AND parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
      BETWEEN parse_datetime('2017-02-18:07:00:00', 'yyyy-MM-dd:HH:mm:ss')
      AND parse_datetime('2017-02-18:08:00:00', 'yyyy-MM-dd:HH:mm:ss');
```

Example – Anzeigen der Menge der in einem festgelegten Zeitraum an eine bestimmten IP-Adresse übertragenen Daten

```
SELECT coalesce(SUM(bytesent), 0) AS bytesenttotal
FROM s3_access_logs_db.mybucket_logs
WHERE remoteip='192.0.2.1'
      AND parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
      BETWEEN parse_datetime('2022-06-01', 'yyyy-MM-dd')
      AND parse_datetime('2022-07-01', 'yyyy-MM-dd');
```

Note

Zur Reduzierung der Beibehaltungsdauer Ihrer Protokolle können Sie für Ihren Serverzugriffsprotokoll-Bucket eine S3-Lebenszykluskonfiguration erstellen. Erstellen Sie Lebenszykluskonfigurationsregeln, um Protokolldateien regelmäßig zu entfernen. Dadurch wird die Menge der Daten reduziert, die Athena in einer Abfrage analysiert. Weitere Informationen finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#).

Identifizieren von Signature-Version-2-Anforderungen mittels Amazon-S3-Zugriffsprotokollen

Die Amazon-S3-Unterstützung für Signature Version 2 wird deaktiviert (veraltet). Danach akzeptiert Amazon S3 keine Anforderungen mit Signature Version 2 mehr, alle Anforderungen müssen also mit Signature Version 4 signiert werden. Sie können Signature-Version-2-Zugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen identifizieren.

Note

Um Signature Version 2-Anfragen zu identifizieren, empfehlen wir die Verwendung AWS CloudTrail Datenereignisse anstelle von Amazon S3 S3-Serverzugriffsprotokollen. CloudTrail Datenereignisse sind einfacher einzurichten und enthalten mehr Informationen als Serverzugriffsprotokolle. Weitere Informationen finden Sie unter [Identifizieren von Amazon S3 Signature Version 2-Anfragen mithilfe von CloudTrail](#).

Example – Anzeigen aller Anforderer, die Signature Version 2-Datenverkehr senden

```
SELECT requester, sigv, Count(sigv) as sigcount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, sigv;
```

Identifizieren von Objektzugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen

Sie können Abfragen an Amazon-S3-Server-Zugriffsprotokolle verwenden, um Amazon-S3-Objektzugriffsanforderungen für Vorgänge zu identifizieren, wie etwa GET, PUT und DELETE, und weitere Informationen über diese Anforderungen zu entdecken.

Das folgende Amazon-Athena-Abfragebeispiel zeigt, wie Sie alle PUT-Objektanforderungen für Amazon S3 aus einem Serverzugriffsprotokoll abrufen können.

Example – Anzeigen aller Anforderer, die **PUT**-Objektanforderungen in einem bestimmten Zeitraum senden

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.PUT.OBJECT' AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

Das folgende Amazon Athena-Abfragebeispiel zeigt, wie alle GET-Objektanfragen für Amazon S3 aus dem Server-Zugriffsprotokoll abgerufen werden.

Example – Anzeigen aller Anforderer, die **GET**-Objektanforderungen in einem bestimmten Zeitraum senden

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.GET.OBJECT' AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

Das folgende Amazon Athena-Abfragebeispiel zeigt, wie alle anonymen Anforderungen aus dem Server-Zugriffsprotokoll in Ihre S3-Buckets gelangen.

Example – Anzeigen aller anonymen Anforderer, die in einem bestimmten Zeitraum Anforderungen an einen Bucket richten

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db.mybucket_logs
WHERE requester IS NULL AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
```

```
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

Die folgende Amazon Athena Athena-Abfrage zeigt, wie Sie alle Anfragen an Ihre S3-Buckets identifizieren können, für die eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich war. Sie können diese Informationen verwenden, um diese ACL Berechtigungen in die entsprechenden Bucket-Richtlinien zu migrieren und zu deaktivieren. ACLs Nachdem Sie diese Bucket-Richtlinien erstellt haben, können Sie sie ACLs für diese Buckets deaktivieren. Weitere Informationen zur Deaktivierung ACLs finden Sie unter. [Voraussetzungen für die Deaktivierung ACLs](#)

Example — Identifizieren Sie alle Anfragen, ACL für die eine Autorisierung erforderlich war

```
SELECT bucket_name, requester, key, operation, aclrequired, requestdatetime
FROM s3_access_logs_db
WHERE aclrequired = 'Yes' AND
parse_datetime(requestdatetime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-05-10:00:00:00', 'yyyy-MM-dd:HH:mm:ss')
AND parse_datetime('2022-08-10:00:00:00', 'yyyy-MM-dd:HH:mm:ss')
```

Note

- Sie können den Datumsbereich nach Belieben an Ihre Anforderungen anpassen.
- Diese Abfragebeispiele können auch für die Sicherheitsüberwachung nützlich sein. Sie können die Ergebnisse auf PutObject- oder GetObject-Aufrufe von unerwarteten oder nicht autorisierten IP-Adressen oder Anforderern und zum Identifizieren anonymer Anforderungen an Ihre Buckets prüfen.
- Diese Abfrage ruft nur Informationen von der Zeit ab, zu der die Protokollierung aktiviert wurde.
- Wenn Sie verwenden AWS CloudTrail Protokolle, siehe [Identifizieren des Zugriffs auf S3-Objekte mithilfe von CloudTrail](#).

Metriken mit Amazon überwachen CloudWatch

CloudWatch Amazon-Metriken für Amazon S3 können Ihnen helfen, die Leistung von Anwendungen, die Amazon S3 verwenden, zu verstehen und zu verbessern. Es gibt mehrere Möglichkeiten, die Sie CloudWatch mit Amazon S3 verwenden können.

Tägliche Speichermetriken für Buckets

Überwachen Sie den Bucket-Speicher mithilfe CloudWatch, der Speicherdaten von Amazon S3 sammelt und zu lesbaren, täglichen Messwerten verarbeitet. Diese Speichermetriken für Amazon S3 werden einmal pro Tag gemeldet und allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt.

Anforderungsmetriken

Überwachen von Amazon-S3-Anforderungen, um Probleme bei der Ausführung schnell zu identifizieren und zu beheben. Die Metriken stehen in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung zur Verfügung. Diese CloudWatch Metriken werden zum gleichen Tarif wie die CloudWatch benutzerdefinierten Amazon-Metriken in Rechnung gestellt. CloudWatch Preisinformationen finden Sie unter [CloudWatch Amazon-Preise](#). Weitere Informationen zum optionalen Erhalt dieser Metriken finden Sie unter [CloudWatch Metriken, Konfigurationen](#).

Wenn die Anforderungsmetriken aktiviert sind, werden sie für alle Objekt-Vorgänge gemeldet. Standardmäßig stehen diese 1-Minuten-Metriken auf Ebene der Amazon-S3-Buckets zur Verfügung. Sie können auch einen Filter für die Metriken definieren, die mit einem gemeinsamen Präfix oder Objekt-Tag oder Zugriffspunkt erfasst wurden.

- Zugriffspunkt – Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind und vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3. Mit dem Zugriffspunkt-Filter erhalten Sie Erkenntnisse in Ihre Zugriffspunkt-Nutzung. Weitere Hinweise zu Zugangspunkten finden Sie unter [Überwachen und Protokollieren von Zugriffspunkten](#).
- Präfix – Obwohl das Amazon-S3-Datenmodell eine flache Struktur aufweist, können Sie die Hierarchie durch die Verwendung eines Präfixes inferieren. Ein Präfix ähnelt einem Verzeichnisnamen, mit dem Sie ähnliche Objekte in einem Bucket gruppieren können. Die S3-Konsole unterstützt Präfixe mit dem Ordnerkonzept. Wenn Sie nach dem Präfix filtern, werden Objekte mit demselben Präfix in die Metrik-Konfiguration aufgenommen. Weitere Informationen zu Präfixen finden Sie unter [Organisieren von Objekten mit Präfixen](#).
- Tag – Tags sind Schlüssel-Wert-Namenspaare, die Sie Objekten hinzufügen können. Mit Markierungen können Sie Objekte einfacher finden und organisieren. Tags können auch als Filter für Metrikkonfigurationen verwendet werden, sodass nur Objekte mit diesen Tags in die Metrikkonfiguration aufgenommen werden. Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).

Um diese Metriken an bestimmten Geschäftsanwendungen, Workflows oder internen Organisationen auszurichten, können Sie nach einem freigegebenen Präfix, einem Objekt-Tag oder einem Zugriffspunkt filtern.

Replikationsmetriken

Replikationsmetriken — Überwachen Sie die Gesamtzahl der API S3-Operationen, für die eine Replikation aussteht, die Gesamtgröße der Objekte, deren Replikation noch aussteht, und die maximale Replikationszeit bis zum Ziel AWS-Region und die Gesamtzahl der Operationen, bei denen die Replikation fehlgeschlagen ist. Replikationsregeln, für die S3 Replication Time Control (S3RTC) oder S3 Replication-Metriken aktiviert sind, veröffentlichen Replikationsmetriken.

Weitere Informationen finden Sie unter [Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status](#) oder [Erfüllung der Compliance-Anforderungen mit S3 Replication Time Control \(S3RTC\)](#).

Metriken von Amazon S3 Storage Lens

Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch [Dashboards](#) zu erstellen. S3-Storage-Lens-Metriken sind im `AWS/S3/Storage-Lens`-Namespace verfügbar. Die CloudWatch Veröffentlichungsoption ist für S3 Storage Lens-Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen aktualisiert wurden. Sie können die CloudWatch Veröffentlichungsoption für eine neue oder bestehende Dashboard-Konfiguration in S3 Storage Lens aktivieren.

Weitere Informationen finden Sie unter [Überwachen von Metriken von S3 Storage Lens in CloudWatch](#).

Alle CloudWatch Statistiken werden für einen Zeitraum von 15 Monaten aufbewahrt, sodass Sie auf historische Informationen zugreifen und sich einen besseren Überblick über die Leistung Ihrer Webanwendung oder Ihres Dienstes verschaffen können. Weitere Informationen zu CloudWatch finden Sie unter [Was ist Amazon CloudWatch?](#) im CloudWatch Amazon-Benutzerhandbuch. Abhängig von Ihren Anwendungsfällen benötigen Sie möglicherweise einige zusätzliche Konfigurationen für Ihre CloudWatch Alarmer. Sie können beispielsweise einen mathematischen Metrik-Ausdruck verwenden, um einen Alarm zu erzeugen. Weitere Informationen finden Sie unter [Verwenden von CloudWatch Metriken](#), [Verwenden von metrischen Berechnungen](#), [Verwenden von CloudWatch Amazon-Alarmen](#) und [Erstellen eines CloudWatch Alarms auf der Grundlage eines metrischen mathematischen Ausdrucks](#) im CloudWatch Amazon-Benutzerhandbuch.

Bereitstellung von CloudWatch Kennzahlen nach bestem Wissen

CloudWatch Die Metriken werden nach bestem Wissen und Gewissen bereitgestellt. Die meisten Anfragen für ein Amazon S3 S3-Objekt mit Anforderungsmetriken führen dazu, dass ein Datenpunkt an gesendet wird CloudWatch.

Die Vollständigkeit und Rechtzeitigkeit der Metriken ist nicht garantiert. Der Datenpunkt für eine bestimmte Anforderung wird möglicherweise mit einem Zeitstempel zurückgegeben, der nach der tatsächlichen Anforderungsverarbeitung liegt. Der Datenpunkt kann sich um eine Minute verzögern, bevor er verfügbar ist CloudWatch, oder er wird möglicherweise gar nicht zugestellt. CloudWatch Anhand von Anforderungsmetriken erhalten Sie nahezu in Echtzeit eine Vorstellung von der Art des Datenverkehrs in Ihrem Bucket. Sie sind nicht als vollständige Abrechnung aller Anforderungen vorgesehen.

Aufgrund der Best-Effort-Natur dieser Funktion enthalten die im [Fakturierungs- und Kostenverwaltungs-Dashboard](#) verfügbaren Berichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in den Bucket-Metriken angezeigt werden.

Weitere Informationen finden Sie unter den folgenden Themen.

Themen

- [Metriken und Dimensionen](#)
- [Zugriff auf CloudWatch Metriken](#)
- [CloudWatch Metriken, Konfigurationen](#)

Metriken und Dimensionen

Die Speichermetriken und Dimensionen, die Amazon S3 an Amazon sendet, CloudWatch sind in den folgenden Tabellen aufgeführt.

Bereitstellung von CloudWatch Kennzahlen nach bestem Wissen

CloudWatch Die Metriken werden nach bestem Wissen und Gewissen bereitgestellt. Die meisten Anfragen für ein Amazon S3 S3-Objekt mit Anforderungsmetriken führen dazu, dass ein Datenpunkt an gesendet wird CloudWatch.

Die Vollständigkeit und Rechtzeitigkeit der Metriken ist nicht garantiert. Der Datenpunkt für eine bestimmte Anforderung wird möglicherweise mit einem Zeitstempel zurückgegeben, der nach der

tatsächlichen Anforderungsverarbeitung liegt. Der Datenpunkt kann sich um eine Minute verzögern, bevor er verfügbar ist CloudWatch, oder er wird möglicherweise gar nicht zugestellt. CloudWatch Anhand von Anforderungsmetriken erhalten Sie nahezu in Echtzeit eine Vorstellung von der Art des Datenverkehrs in Ihrem Bucket. Sie sind nicht als vollständige Abrechnung aller Anforderungen vorgesehen.

Aufgrund der Best-Effort-Natur dieser Funktion enthalten die im [Fakturierungs- und Kostenverwaltungs-Dashboard](#) verfügbaren Berichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in den Bucket-Metriken angezeigt werden.


Themen

- [Tägliche Amazon S3 S3-Speichermetriken für Buckets in CloudWatch](#)
- [Amazon S3 S3-Anforderungsmetriken in CloudWatch](#)
- [Metriken zur S3-Replikation in CloudWatch](#)
- [S3 Storage Lens-Metriken in CloudWatch](#)
- [Lambda-Anforderungsmetriken für S3-Objekte in CloudWatch](#)
- [Amazon S3 auf Outposts — Metriken in CloudWatch](#)
- [Amazon S3 S3-Abmessungen in CloudWatch](#)
- [Dimensionen der S3-Replikation in CloudWatch](#)
- [Abmessungen der S3-Speicherlinse in CloudWatch](#)
- [Abmessungen der Lambda-Anforderung von S3 Object in CloudWatch](#)

Tägliche Amazon S3 S3-Speichermetriken für Buckets in CloudWatch

Der AWS/S3-Namespace enthält die folgenden täglichen Speichermetriken für Buckets.

Metrik	Beschreibung
BucketSizeBytes	<p>Die Datenmenge in Byte, die in einem Bucket in den folgenden Speicherklassen gespeichert wird:</p> <ul style="list-style-type: none"> • S3 Standard (STANDARD) • S3 Intelligent-Tiering (INTELLIGENT_TIERING) • S3 Standard-Infrequent Access (STANDARD_IA) • S3 One Zone — Seltener Zugriff () ONEZONE_IA

Metrik	Beschreibung
	<ul style="list-style-type: none"> • Speicher mit reduzierter Redundanz () () RRS REDUCED_REDUNDANCY • S3 Glacier Instant Retrieval (GLACIER_IR) • S3 Glacier Deep Archive (DEEP_ARCHIVE) • S3 Glacier Flexible Retrieval (GLACIER) • S3 Express One Zone (EXPRESS_ONEZONE) <p>Dieser Wert wird berechnet, indem die Größe aller Objekte und Metadaten (wie Bucket-Namen) im Bucket (sowohl aktuelle als auch nicht aktuelle Objekte) summiert wird, einschließlich der Größe aller Teile für alle unvollständigen mehrteiligen Uploads in den Bucket.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Die S3 Express One Zone-Speicherklasse ist nur für Verzeichnis-Buckets verfügbar.</p> </div> <p>Gültige Speichertypfilter (siehe Dimension): StorageType</p> <ul style="list-style-type: none"> • S3-Standard: StandardStorage • Intelligentes S3-Tiering: IntelligentTieringFAStorage ,,, IntelligentTieringIAStorage IntelligentTieringAAStorage IntelligentTieringAIASStorage IntelligenteTieringDAASStorage • S3 Standard — Seltener Zugriff:,, StandardIAStorage StandardIASizeOverhead StandardIAObjectOverhead • S3 One Zone — Seltener Zugriff:, OneZoneIAStorage OneZoneIASizeOverhead • Speicher mit reduzierter Redundanz (): RRS ReducedRedundancyStorage • S3 Glacier Sofortiger Abruf:, GlacierIRSizeOverhead GlacierInstantRetrievalStorage


Metrik	Beschreibung
	<ul style="list-style-type: none"> • Flexibler S3 Glacier-Abruf: GlacierStorage ,, GlacierStagingStorage GlacierObjectOverhead GlacierS3ObjectOverhead • S3 Glacier Deep Archive: DeepArchiveStorage ,DeepArchiveObjectOverhead ,DeepArchiveS3ObjectOverhead , DeepArchiveStagingStorage • S3 Express Eine Zone: ExpressOneZoneStorage <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt</p> <p>Weitere Informationen zu den StorageType Abmessungen finden Sie unter the section called “Amazon S3 S3-Abmessungen in CloudWatch”.</p>
NumberOfObjects	<p>Die Gesamtzahl der Objekte für alle Speicherklassen, die in einem Bucket für allgemeine Zwecke gespeichert sind. Zur Berechnung dieses Werts werden alle aktuellen und nicht aktuellen Objekte im Bucket, alle Löschmarkierungen sowie die Gesamtanzahl der Teile sämtlicher unvollständiger mehrteiliger Uploads in den Bucket gezählt. Bei Verzeichnis-Buckets mit Objekten in der S3 Express One Zone-Speicherkategorie wird dieser Wert berechnet, indem alle Objekte im Bucket gezählt werden. Unvollständige Mehrfach-Uploads in den Bucket werden jedoch nicht berücksichtigt.</p> <p>Gültige Speichertypfilter: nur AllStorageTypes (siehe StorageType -Dimension)</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt</p>


Amazon S3 S3-Anforderungsmetriken in CloudWatch

Der AWS/S3-Namespace enthält die folgenden Anforderungsmetriken. Zu diesen Metriken gehören Anfragen, die nicht abgerechnet werden können (im Fall von GET Anfragen von CopyObject und Replikation).

Note

Amazon S3 S3-Anforderungsmetriken in werden für Directory-Buckets CloudWatch nicht unterstützt.

Metrik	Beschreibung
AllRequests	<p>Die Gesamtzahl der HTTP Anfragen an einen Amazon S3 S3-Bucket , unabhängig vom Typ. Wenn Sie eine Metrikkonfiguration mit einem Filter verwenden, gibt diese Metrik nur die HTTP Anfragen zurück, die die Anforderungen des Filters erfüllen.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
GetRequests	<p>Die Anzahl der HTTP GET Anfragen für Objekte in einem Amazon S3 S3-Bucket. Das Auflisten von Vorgängen ist hierin nicht inbegriffen. Diese Metrik wird für die Quelle jeder CopyObject Anfrage inkrementiert.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p> <div data-bbox="500 1570 623 1608" data-label="Section-Header"> <h3> Note</h3> </div> <div data-bbox="544 1623 1450 1757" data-label="Text"> <p>Paginierte, listenorientierte Anfragen, wie ListMultipartUploads, ListParts, ListObjectVersions und andere sind in dieser Metrik nicht enthalten.</p> </div>

Metrik	Beschreibung
PutRequests	<p>Die Anzahl der HTTP PUT Anfragen für Objekte in einem Amazon S3 S3-Bucket. Diese Metrik wird für das Ziel jeder CopyObject Anfrage inkrementiert.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
DeleteRequests	<p>Die Anzahl der HTTP DELETE Anfragen für Objekte in einem Amazon S3 S3-Bucket. Diese Metrik umfasst auch DeleteObjectsAnfragen. Diese Metrik zeigt die Anzahl der gestellten Anfragen, nicht die Anzahl der gelöschten Objekte an.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
HeadRequests	<p>Die Anzahl der HTTP HEAD Anfragen an einen Amazon S3 S3-Bucket.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
PostRequests	<p>Die Anzahl der HTTP POST Anfragen an einen Amazon S3 S3-Bucket.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p> <div data-bbox="472 1423 1507 1642"><p> Note</p><p>DeleteObjects und SelectObjectContentAnfragen sind in dieser Metrik nicht enthalten.</p></div>

Metrik	Beschreibung
SelectRequests	<p>Die Anzahl der Amazon S3 SelectObjectContentAnfragen für Objekte in einem Amazon S3 S3-Bucket.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
SelectBytesScanned	<p>Die Anzahl der mit Amazon S3 gescannten Datenbytes SelectObjectContentAnfragen in einem Amazon S3 S3-Bucket.</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9</p>
SelectBytesReturned	<p>Die Anzahl der mit Amazon S3 zurückgegebenen Datenbytes SelectObjectContentAnfragen in einem Amazon S3 S3-Bucket.</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9</p>
ListRequests	<p>Die Anzahl der HTTP Anfragen, die den Inhalt eines Buckets auflisten.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
BytesDownloaded	<p>Anzahl der heruntergeladenen Bytes für Anfragen an einen Amazon-S3-Bucket, wobei die Antwort einen Textkörper enthält.</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9</p>

Metrik	Beschreibung
BytesUploaded	<p>Anzahl der hochgeladenen Bytes für Anfragen an einen Amazon-S3-Bucket, wobei die Antwort einen Textkörper enthält.</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9</p>
4xxErrors	<p>Die Anzahl der HTTP 4 x x Client-Fehlerstatuscode-Anfragen, die an einen Amazon S3 S3-Bucket mit einem Wert von entweder 0 oder 1 gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl</p>
5xxErrors	<p>Die Anzahl von HTTP 5 xx-Serverfehlerstatuscode-Anfragen, die an einen Amazon S3 S3-Bucket mit einem Wert von entweder 0 oder 1 gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl</p>

Metrik	Beschreibung
FirstByteLatency	<p>Zeit pro Anforderung vom Zeitpunkt des Eingangs der vollständigen Anfragen bei einem Amazon-S3-Bucket bis das Zurückgeben der Antwort beginnt.</p> <p>Einheiten: Millisekunden</p> <p>Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100</p>
TotalRequestLatency	<p>Zeit pro Anforderung vom Zeitpunkt des Eingangs des ersten Bytes bei einem Amazon-S3-Bucket bis zum Senden des letzten Bytes. Diese Metrik umfasst die erforderliche Zeit, um den Textkörper zu erhalten und den Antworttextkörper zu senden, welche nicht in FirstByteLatency enthalten ist.</p> <p>Einheiten: Millisekunden</p> <p>Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100</p>

Metriken zur S3-Replikation in CloudWatch

Sie können den Fortschritt der Replikation mit S3-Replikationsmetriken überwachen, indem Sie ausstehende Bytes, ausstehende Operationen und die Replikationslatenz nachverfolgen. Weitere Informationen finden Sie unter [Überwachen des Fortschritts mit Replikationsmetriken](#).

Note

Sie können Alarme für Ihre Replikationsmetriken in Amazon aktivieren CloudWatch. Wenn Sie Alarme für Ihre Replikationsmetriken einrichten, stellen Sie das Feld Missing data treatment (Behandlung fehlender Daten) auf Treat missing data as ignore (maintain the alarm state) (Fehlende Daten ignorieren) (Alarmstatus beibehalten) ein.

Metrik	Beschreibung
ReplicationLatency	<p>Die maximale Anzahl von Sekunden, um die das Replikationsziel AWS-Region befindet sich hinter der Quelle AWS-Region für eine bestimmte Replikationsregel.</p> <p>Einheiten: Sekunden</p> <p>Gültige Statistiken: Max</p>
BytesPendingReplication	<p>Die Gesamtanzahl der Bytes von Objekten, die für eine bestimmte Replikationsregel ausstehen.</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Max</p>
OperationsPendingReplication	<p>Die Anzahl der Vorgänge mit ausstehender Replikation für eine bestimmte Replikationsregel.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Max</p>
OperationsFailedReplication	<p>Die Anzahl der Operationen, deren Replikation für eine bestimmte Replikationsregel fehlgeschlagen ist.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe (Gesamtzahl der fehlgeschlagenen Operationen), Durchschnitt (Fehlerrate), Stichprobenanzahl (Gesamtzahl der Replikationsoperationen)</p>

S3 Storage Lens-Metriken in CloudWatch

Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in [CloudWatchDashboards](#) zu erstellen. S3 Storage Lens-Metriken werden im AWS/S3/Storage-Lens Namespace in

veröffentlicht. CloudWatch Die CloudWatch Veröffentlichungsoption ist für S3 Storage Lens-Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen aktualisiert wurden.

Eine Liste der S3 Storage Lens-Metriken, die veröffentlicht wurden CloudWatch, finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#). Eine vollständige Liste der Dimensionen finden Sie unter [Dimensionen](#).

Lambda-Anforderungsmetriken für S3-Objekte in CloudWatch

S3 Objekt Lambda enthält die folgenden Anforderungsmetriken.

Metrik	Beschreibung
AllRequests	<p>Die Gesamtzahl der HTTP Anfragen, die mithilfe eines Object Lambda Access Points an einen Amazon S3 S3-Bucket gestellt wurden.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
GetRequests	<p>Die Anzahl der HTTP GET Anfragen für Objekte, die mithilfe eines Object Lambda Access Points gestellt wurden. Diese Metrik umfasst keine Operationen zum Auflisten.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
BytesUploaded	<p>Die Anzahl von Bytes, die mithilfe eines Object Lambda Access Point in einen Amazon-S3-Bucket hochgeladen wurden, wobei die Anforderung einen Textkörper enthält.</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9</p>
PostRequests	<p>Die Anzahl der HTTP POST Anfragen, die mithilfe eines Object Lambda Access Points an einen Amazon S3 S3-Bucket gestellt wurden.</p>

Metrik	Beschreibung
	Einheiten: Anzahl Gültige Statistiken: Summe
PutRequests	Die Anzahl der HTTP PUT Anfragen für Objekte in einem Amazon S3 S3-Bucket, die mithilfe eines Object Lambda Access Points gestellt wurden. Einheiten: Anzahl Gültige Statistiken: Summe
DeleteRequests	Die Anzahl der HTTP DELETE Anfragen für Objekte in einem Amazon S3 S3-Bucket, die mithilfe eines Object Lambda Access Points gestellt wurden. Diese Metrik umfasst DeleteObjects Anfragen. Diese Metrik zeigt die Anzahl der gestellten Anfragen, nicht die Anzahl der gelöschten Objekte an. Einheiten: Anzahl Gültige Statistiken: Summe
BytesDownloaded	Die Anzahl der heruntergeladenen Bytes für Anforderungen an einen Amazon-S3-Bucket mithilfe eines Object Lambda Access Point, wobei die Antwort einen Textkörper enthält Einheiten: Byte Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9

Metrik	Beschreibung
FirstByteLatency	<p>Zeit pro Anforderung vom Zeitpunkt des Eingangs der vollständigen Anfragen bei einem Amazon-S3-Bucket über einen Object Lambda Access Point bis zum Beginn des Zurückgebens der Antwort. Diese Metrik ist abhängig von AWS Lambda die Laufzeit der Funktion zur Transformation des Objekts, bevor die Funktion die Bytes an den Object Lambda Access Point zurückgibt.</p> <p>Einheiten: Millisekunden</p> <p>Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100</p>
TotalRequestLatency	<p>Zeit pro Anforderung vom Zeitpunkt des Eingangs des ersten Bytes bis zum Senden des letzten Bytes an einen Object Lambda Access Point. Diese Metrik umfasst die erforderliche Zeit, um den Textkörper zu erhalten und den Antworttextkörper zu senden, welche nicht in FirstByteLatency enthalten ist.</p> <p>Einheiten: Millisekunden</p> <p>Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100</p>
HeadRequests	<p>Die Anzahl der HTTP HEAD Anfragen, die mithilfe eines Object Lambda Access Points an einen Amazon S3 S3-Bucket gestellt wurden.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
ListRequests	<p>Die Anzahl der HTTP GET Anfragen, die den Inhalt eines Amazon S3 S3-Buckets auflisten. Diese Metrik enthält sowohl ListObjects - als auch ListObjectsV2 -Operationen.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>

Metrik	Beschreibung
4xxErrors	<p>Die Anzahl der HTTP 4 xx Client-Fehlerstatuscode-Anfragen, die mithilfe eines Object Lambda Access Points mit einem Wert von entweder 0 oder 1 an einen Amazon S3 S3-Bucket gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl</p>
5xxErrors	<p>Die Anzahl der HTTP 5 xx-Serverfehlerstatuscode-Anfragen, die mithilfe eines Object Lambda Access Points mit einem Wert von entweder 0 oder 1 an einen Amazon S3 S3-Bucket gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl</p>
ProxiedRequests	<p>Die Anzahl der HTTP Anfragen an einen Object Lambda Access Point, die die API Standardantwort von Amazon S3 zurückgeben. (Für solche Anforderungen ist keine Lambda-Funktion konfiguriert.)</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
InvokedLambda	<p>Die Anzahl der HTTP Anfragen an ein S3-Objekt, bei denen eine Lambda-Funktion aufgerufen wurde.</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>

Metrik	Beschreibung
LambdaResponseRequests	Die Anzahl von WriteGetObjectResponse -Anforderungen, die von der Lambda-Funktion gestellt wurden. Diese Metrik gilt nur für GetObject -Anforderungen.
LambdaResponse4xx	Die Anzahl von HTTP 4xx Client-Fehlern, die beim Aufrufen WriteGetObjectResponse von einer Lambda-Funktion aus auftreten. Diese Metrik liefert dieselben Informationen wie 4xxErrors , jedoch nur für WriteGetObjectResponse -Aufrufe.
LambdaResponse5xx	Die Anzahl der HTTP 5xx-Serverfehler, die beim Aufrufen WriteGetObjectResponse von einer Lambda-Funktion auftreten. Diese Metrik liefert dieselben Informationen wie 5xxErrors , jedoch nur für WriteGetObjectResponse -Aufrufe.

Amazon S3 auf Outposts — Metriken in CloudWatch

Eine Liste der Metriken CloudWatch , die für S3 in Outposts-Buckets verwendet werden, finden Sie unter. [CloudWatch Metriken](#)

Amazon S3 S3-Abmessungen in CloudWatch

Die nachstehenden Dimensionen werden verwendet, um Amazon-S3-Metriken zu filtern.

Dimension	Beschreibung
BucketName	Diese Dimension filtert die angeforderten Daten nur für den identifizierten Bucket.
StorageType	Diese Dimension filtert die Daten, die Sie in einem Bucket gespeichert haben, nach den folgenden Speichertypen: <ul style="list-style-type: none"> StandardStorage – Die Gesamtzahl der Bytes für Objekte in der STANDARD-Speicherklasse. IntelligentTieringAAStorage – Die Anzahl von Bytes für Objekte in Archive Access Tier der INTELLIGENT_TIERING -Speicherklasse.

Dimension	Beschreibung
	<ul style="list-style-type: none"> • <code>IntelligentTieringAIASStorage</code> – Die Anzahl von Bytes für Objekte in Stufe Archive Instant Access Tier der <code>INTELLIGENT_TIERING</code> -Speicherklasse. • <code>IntelligentTieringDAASStorage</code> – Die Anzahl von Bytes für Objekte in Deep Archive Access Tier der <code>INTELLIGENT_TIERING</code> -Speicherklasse. • <code>IntelligentTieringFASStorage</code> – Die Anzahl von Bytes für Objekte in Frequent Access Tier der <code>INTELLIGENT_TIERING</code> -Speicherklasse. • <code>IntelligentTieringIASStorage</code> – Die Anzahl von Bytes für Objekte in Infrequent Access Tier der <code>INTELLIGENT_TIERING</code> -Speicherklasse. • <code>StandardIASStorage</code> — Die Anzahl der Byte, die für Objekte in der Speicherklasse S3 Standard-Infrequent Access (<code>STANDARD_IA</code>) verwendet werden. • <code>StandardIASizeOverhead</code> – Die Anzahl von Bytes für Objekte mit einer Größe von weniger als 128 KB in der <code>STANDARD_IA</code> -Speicherklasse. • <code>IntAAObjectOverhead</code> – Für jedes Objekt in der <code>INTELLIGENT_TIERING</code> -Speicherklasse in Archive Access Tier fügt S3 Glacier 32 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die Gebühren für S3 Glacier Flexible Retrieval in Rechnung gestellt. • <code>IntAAS3ObjectOverhead</code> – Für jedes Objekt in der <code>INTELLIGENT_TIERING</code> -Speicherklasse in Archive Access Tier verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt.

Dimension	Beschreibung
	<ul style="list-style-type: none"> <li data-bbox="591 212 1507 579">• <code>IntDAAObjectOverhead</code> – Für jedes Objekt in der <code>INTELLIGENT_TIERING</code> -Speicherklasse in Deep Archive Access Tier fügt S3 Glacier 32 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die S3 Glacier Deep Archive-Speichergebühren in Rechnung gestellt. <li data-bbox="591 604 1507 926">• <code>IntDAAS3ObjectOverhead</code> – Für jedes Objekt in der <code>INTELLIGENT_TIERING</code> -Speicherklasse in Deep Archive Access Tier fügt Amazon S3 8 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt. <li data-bbox="591 951 1451 1079">• <code>OneZoneIASStorage</code> – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 One Zone-Infrequent Access (<code>ONEZONE_IA</code>). <li data-bbox="591 1104 1451 1232">• <code>OneZoneIASizeOverhead</code> – Die Anzahl von Bytes für Objekte mit einer Größe von weniger als 128 KB in der <code>ONEZONE_IA</code> -Speicherklasse. <li data-bbox="591 1257 1500 1386">• <code>ReducedRedundancyStorage</code> — Die Anzahl der Byte, die für Objekte in der Klasse Reduced Redundancy Storage () verwendet werden. RRS <li data-bbox="591 1411 1500 1539">• <code>GlacierIRSizeOverhead</code> – Die Anzahl von Bytes für Objekte, die kleiner als 128 KB sind, in der Speicherklasse S3 Glacier Instant Retrieval. <li data-bbox="591 1564 1500 1692">• <code>GlacierInstantRetrievalStorage</code> – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 Glacier Instant Retrieval. <li data-bbox="591 1717 1500 1797">• <code>GlacierStorage</code> – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval.

Dimension	Beschreibung
	<ul style="list-style-type: none"> <li data-bbox="592 212 1479 436">• <code>GlacierStagingStorage</code> – Die Anzahl von Bytes für Teile von mehrteiligen Objekten, bevor die <code>CompleteMultipartUpload</code> -Anforderung für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval abgeschlossen ist. <li data-bbox="592 464 1490 779">• <code>GlacierObjectOverhead</code> – Für jedes archivierte Objekt fügt S3 Glacier 32 KB Speicher für den Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die Gebühren für S3 Glacier Flexible Retrieval in Rechnung gestellt. <li data-bbox="592 806 1463 1031">• <code>GlacierS3ObjectOverhead</code> – Für jedes in S3 Glacier Flexible Retrieval archivierte Objekt verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt. <li data-bbox="592 1058 1409 1136">• <code>DeepArchiveStorage</code> – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 Glacier Deep Archive. <li data-bbox="592 1163 1507 1430">• <code>DeepArchiveObjectOverhead</code> – Für jedes archivierte Objekt fügt S3 Glacier 32 KB Speicher für den Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die S3 Glacier Deep Archive-Gebühren in Rechnung gestellt. <li data-bbox="592 1457 1463 1682">• <code>DeepArchiveS3ObjectOverhead</code> – Für jedes in S3 Glacier Deep Archive archivierte Objekt verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt. <li data-bbox="592 1709 1479 1787">• <code>DeepArchiveStagingStorage</code> – Die Anzahl von Bytes für Teile von mehrteiligen Objekten, bevor die <code>CompleteM</code>

Dimension	Beschreibung
	<ul style="list-style-type: none"> • <code>multipartUpload</code> – Anforderung für Objekte in der Speicherklasse S3 Glacier Deep Archive abgeschlossen ist. • <code>ExpressOneZoneStorage</code> – Die Gesamtzahl der Bytes für Objekte in der S3-Express-One-Zone-Speicherklasse.
<code>FilterId</code>	Diese Dimension filtert Metrikkonfigurationen, die Sie für Anforderungsmetriken für einen Bucket angeben. Wenn Sie eine Metrikkonfiguration erstellen, geben Sie eine Filter-ID an (z. B. ein Präfix, ein Tag oder einen Zugriffspunkt). Weitere Informationen finden Sie unter Erstellen einer Metrik-Konfiguration .

Dimensionen der S3-Replikation in CloudWatch

Die folgenden Dimensionen werden verwendet, um S3-Replikationsmetriken zu filtern.

Dimension	Beschreibung
<code>SourceBucket</code>	Der Name der Bucket-Objekte, aus denen repliziert wird.
<code>DestinationBucket</code>	Der Name der Bucket-Objekte, in die repliziert werden.
<code>RuleId</code>	Eine eindeutige Kennung für die Regel, die die Aktualisierung dieser Replikationsmetrik ausgelöst hat.

Abmessungen der S3-Speicherlinse in CloudWatch

Eine Liste der Dimensionen, die zum Filtern von S3 Storage Lens-Metriken verwendet werden CloudWatch, finden Sie unter [Dimensionen](#).

Abmessungen der Lambda-Anforderung von S3 Object in CloudWatch

Die folgenden Dimensionen werden verwendet, um Daten aus einem Object Lambda Access Point zu filtern.

Dimension	Beschreibung
AccessPointName	Der Name des Zugriffspunkts, von dem Anforderungen gestellt werden.
DataSourceARN	Die Quelle, aus der der Object Lambda Access Point die Daten abrufen. Wenn die Anfrage eine Lambda-Funktion aufruft, bezieht sich diese auf den Lambda-Amazon-Ressourcennamen (). ARN Andernfalls bezieht sich dies auf den Access Point. ARN

Zugriff auf CloudWatch Metriken

Sie können die folgenden Vorgehensweisen nutzen, um die Speichermetriken von Amazon S3 anzuzeigen. Damit Amazon-S3-Metriken berücksichtigt werden, müssen Sie Zeitstempel für Anfang und Ende angeben. Für Metriken für einen bestimmten 24-Stunden-Zeitraum setzen Sie das Zeitintervall auf 86400 Sekunden, die Anzahl der Sekunden eines Tages. Denken Sie daran, die Dimensionen BucketName und StorageType festzulegen.

Verwendung der AWS CLI

Zum Beispiel, wenn Sie die verwenden möchten AWS CLI Um den Durchschnitt der Größe eines bestimmten Buckets in Byte zu ermitteln, könnten Sie den folgenden Befehl verwenden:

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --namespace AWS/S3
--start-time 2016-10-19T00:00:00Z --end-time 2016-10-20T00:00:00Z --statistics Average
--unit Bytes --region us-west-2 --dimensions Name=BucketName,Value=amzn-s3-demo-bucket
Name=StorageType,Value=StandardStorage --period 86400 --output json
```

Dieses Beispiel erzeugt die folgende Ausgabe.

```
{
  "Datapoints": [
    {
      "Timestamp": "2016-10-19T00:00:00Z",
      "Average": 1025328.0,
      "Unit": "Bytes"
    }
  ],
  "Label": "BucketSizeBytes"
```

```
}
```

Verwenden der S3-Konsole

So zeigen Sie Metriken mit der CloudWatch Amazon-Konsole an

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im linken Navigationsbereich Metrics (Metriken) aus.
3. Wählen Sie den Namespace S3 aus.
4. (Optional) Um eine Metrik anzuzeigen, geben Sie den Metriknamen in das Suchfeld ein.
5. (Optional) Um nach der StorageTypeDimension zu filtern, geben Sie den Namen der Speicherklasse in das Suchfeld ein.

Um eine Liste gültiger Messwerte anzuzeigen, die für Ihr gespeichert sind AWS-Konto mithilfe der AWS CLI

- Geben Sie als Eingabeaufforderung den folgenden Befehl ein.

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

Weitere Informationen zu den für den Zugriff auf CloudWatch Dashboards erforderlichen Berechtigungen finden Sie unter [CloudWatch Amazon-Dashboard-Berechtigungen](#) im CloudWatch Amazon-Benutzerhandbuch.

CloudWatch Metriken, Konfigurationen

Mit Amazon CloudWatch Request Metrics für Amazon S3 können Sie CloudWatch 1-Minuten-Metriken erhalten, CloudWatch Alarmer einrichten und auf CloudWatch Dashboards zugreifen, um den near-real-time Betrieb und die Leistung Ihres Amazon S3 S3-Speichers zu überprüfen. Für Anwendungen, die von Cloud-Speicher abhängig sind, ermöglichen diese Metriken Ihnen, Betriebsprobleme schnell zu identifizieren und entsprechende Maßnahmen zu ergreifen. Wenn diese 1-Minuten-Metriken aktiviert sind, stehen sie standardmäßig auf Ebene der Amazon-S3-Buckets zur Verfügung.

Wenn Sie die CloudWatch Anforderungsmetriken für die Objekte in einem Bucket abrufen möchten, müssen Sie eine Metrikkonfiguration für den Bucket erstellen. Weitere Informationen finden Sie unter [Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket](#).

Sie können auch ein freigegebenes Präfix, Objekt-Tags oder einen Zugriffspunkt verwenden, um einen Filter für die gesammelten Metriken zu definieren. Mit dieser Methode zum Definieren eines Filters können Sie also Metrikfilter an bestimmte Geschäftsanwendungen, Workflows oder interne Organisationen anpassen. Weitere Informationen finden Sie unter [Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert](#). Weitere Informationen zu den verfügbaren CloudWatch Metriken und den Unterschieden zwischen Speicher- und Anforderungsmetriken finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

Berücksichtigen Sie bei Verwendung von Metrik-Konfigurationen Folgendes:

- Sie können maximal 1 000 Metrik-Konfigurationen pro Bucket verwenden.
- Unter Verwendung von Filtern können Sie wählen, welche Objekte in einem Bucket Metrik-Konfigurationen enthalten sollen. Sie können nach einem freigegebenen Präfix, Objekt-Tag oder Zugriffspunkt filtern, um Metrikfilter an bestimmte Geschäftsanwendungen, Workflows oder interne Organisationen auszurichten. Um Metriken für den gesamten Bucket anzufordern, erstellen Sie eine Metrik-Konfiguration ohne Filter.
- Metrik-Konfigurationen sind nur erforderlich, um Anforderungsmetriken zu aktivieren. Tägliche Speichermetriken auf Bucket-Ebene sind immer aktiviert und werden allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt. Derzeit ist es nicht möglich, tägliche Speichermetriken für eine gefilterte Untermenge von Objekten zu erhalten.
- Jede Metrik-Konfiguration unterstützt den gesamten Satz [verfügbarer Anforderungsmetriken](#). Operationsspezifische Metriken (z. B. `PostRequests`) werden nur gemeldet, wenn es Anforderungen dieses Typs für den Bucket oder Filter gibt.
- Anforderungsmetriken werden für Vorgänge auf Objektebene gemeldet. Sie werden auch für Operationen gemeldet, die Bucket-Inhalte auflisten, wie [GETBucket \(List Objects\)](#), [GETBucket Object Versions](#) und [List Multipart Uploads](#), aber sie werden nicht für andere Operationen mit Buckets gemeldet.
- Anforderungsmetriken unterstützen das Filtern mit Präfixen, Objekt-Tags oder Zugriffspunkten, Speichermetriken dagegen nicht.

Bereitstellung von Kennzahlen nach bestem Wissen CloudWatch

CloudWatch Die Metriken werden nach bestem Wissen und Gewissen bereitgestellt. Die meisten Anfragen für ein Amazon S3 S3-Objekt mit Anforderungsmetriken führen dazu, dass ein Datenpunkt an gesendet wird CloudWatch.

Die Vollständigkeit und Rechtzeitigkeit der Metriken ist nicht garantiert. Der Datenpunkt für eine bestimmte Anforderung wird möglicherweise mit einem Zeitstempel zurückgegeben, der nach der tatsächlichen Anforderungsverarbeitung liegt. Der Datenpunkt kann sich um eine Minute verzögern, bevor er verfügbar ist CloudWatch, oder er wird möglicherweise gar nicht zugestellt. CloudWatch Anhand von Anforderungsmetriken erhalten Sie nahezu in Echtzeit eine Vorstellung von der Art des Datenverkehrs in Ihrem Bucket. Sie sind nicht als vollständige Abrechnung aller Anforderungen vorgesehen.

Aufgrund der Best-Effort-Natur dieser Funktion enthalten die im [Fakturierungs- und Kostenverwaltungs-Dashboard](#) verfügbaren Berichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in den Bucket-Metriken angezeigt werden.

Weitere Informationen zur Arbeit mit CloudWatch Metriken in Amazon S3 finden Sie in den folgenden Themen.

Themen

- [Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket](#)
- [Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert](#)
- [Löschen eines Metrikfilters](#)

Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket

Wenn Sie Anforderungsmetriken konfigurieren, können Sie eine CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket erstellen oder nach Präfix, Objekt-Tag oder Access Point filtern. Die Verfahren in diesem Thema zeigen Ihnen, wie Sie eine Konfiguration für alle Objekte in Ihrem Bucket erstellen. Informationen zum Erstellen einer Konfiguration, die nach Objekt-Tag, Präfix oder Zugriffspunkt filtert, finden Sie unter [Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert](#).

Es gibt drei Arten von CloudWatch Amazon-Metriken für Amazon S3: Speichermetriken, Anforderungsmetriken und Replikationsmetriken. Speichermetriken werden einmal pro Tag gemeldet und allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt. Die Anforderungsmetriken sind in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung verfügbar. Anforderungsmetriken werden zum CloudWatch Standardtarif abgerechnet. Sie müssen sich dafür entscheiden, Metriken anzufordern, indem Sie sie in der Konsole konfigurieren oder Amazon S3 verwendenAPI. [S3-Replikationsmetriken](#) enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Mit Replikationsmetriken können Sie den minute-by-minute Fortschritt

überwachen, indem Sie ausstehende Bytes, ausstehende Operationen, Operationen, bei denen die Replikation fehlgeschlagen ist, und die Replikationslatenz verfolgen.

Weitere Informationen zu CloudWatch Metriken für Amazon S3 finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

Sie können Metrikkonfigurationen zu einem Bucket hinzufügen, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI) oder das Amazon S3 RESTAPI.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen des Buckets mit den Objekte aus, für die Sie Anforderungsmetriken anfordern möchten.
3. Wählen Sie den Tab Metrics (Metriken).
4. Wählen Sie unter Bucket metrics (Bucket-Metriken) View additional charts (Zusätzliche Diagramme anzeigen) aus.
5. Wählen Sie den Tab Request metrics (Anforderungsmetriken).
6. Wählen Sie Create Filter (Filter erstellen).
7. Geben Sie im Feld Filter name (Filtername) Ihren Filternamen ein.

Namen dürfen nur Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Wir empfehlen, den Namen EntireBucket für einen Filter zu verwenden, der für alle Objekte gilt.

8. Wählen Sie unter Filter scope (Filterbereich) die Option This filter applies to all objects in the bucket (Dieser Filter gilt für alle Objekte im Bucket).

Sie können auch einen Filter definieren, sodass die Metriken nur für eine Untermenge von Objekten im Bucket erfasst und gemeldet werden. Weitere Informationen finden Sie unter [Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert](#).

9. Wählen Sie Save Changes (Änderungen speichern).
10. Wählen Sie im Tab Request metrics (Anforderungsmetriken) unter Filters (Filter) den Filter aus, den Sie gerade erstellt haben.

Nach etwa 15 Minuten CloudWatch beginnt die Erfassung dieser Anforderungsmetriken. Sie können sie im Tab Request metrics (Anforderungsmetriken) sehen. Sie können Diagramme

für die Metriken auf Amazon S3 oder der CloudWatch Konsole sehen. Anforderungsmetriken werden zum CloudWatch Standardtarif in Rechnung gestellt. Weitere Informationen finden Sie unter [CloudWatch Amazon-Preise](#).

Mit dem REST API

Sie können Metrikkonfigurationen auch programmgesteuert mit dem Amazon S3 hinzufügen. REST API Weitere Informationen zum Hinzufügen und Arbeiten mit Metrikkonfigurationen finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference:

- [PUTKonfiguration von Bucket-Metriken](#)
- [GETKonfiguration von Bucket Metric](#)
- [List Bucket-Metrikkonfiguration](#)
- [DELETEKonfiguration von Bucket Metric](#)

Verwendung der AWS CLI

1. Installieren und einrichten Sie das AWS CLI. Anweisungen finden Sie unter [Installation, Aktualisierung und Deinstallation von AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.
2. Öffnen Sie ein Terminalfenster.
3. Führen Sie den folgenden Befehl aus, um eine Metrik-Konfiguration hinzuzufügen.

```
aws s3api put-bucket-metrics-configuration --endpoint https://s3.us-west-2.amazonaws.com --bucket bucket-name --id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id"}'
```

Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert

Es gibt drei Arten von CloudWatch Amazon-Metriken für Amazon S3: Speichermetriken, Anforderungsmetriken und Replikationsmetriken. Speichermetriken werden einmal pro Tag gemeldet und allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt. Die Anforderungsmetriken sind in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung verfügbar. Anforderungsmetriken werden zum CloudWatch Standardtarif abgerechnet. Sie müssen sich dafür entscheiden, Metriken anzufordern, indem Sie sie in der Konsole konfigurieren oder Amazon S3

verwendenAPI. [S3-Replikationsmetriken](#) enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Mit Replikationsmetriken können Sie den minute-by-minute Fortschritt überwachen, indem Sie ausstehende Bytes, ausstehende Operationen, Operationen, bei denen die Replikation fehlgeschlagen ist, und die Replikationslatenz verfolgen.

Weitere Informationen zu CloudWatch Metriken für Amazon S3 finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).

Wenn Sie CloudWatch Metriken konfigurieren, können Sie einen Filter für alle Objekte in Ihrem Bucket erstellen, oder Sie können die Konfiguration in Gruppen verwandter Objekte innerhalb eines einzelnen Buckets filtern. Sie können Objekte in einem Bucket zur Aufnahme in eine Metrik-Konfiguration abhängig von einem oder mehreren der folgenden Filtertypen filtern:

- Object key name prefix (Objekt-Schlüsselnamepräfix) – Obwohl das Amazon-S3-Datenmodell eine flache Struktur aufweist, können Sie eine Hierarchie durch die Verwendung eines Präfixes inferieren. Die Amazon-S3-Konsole unterstützt diese Präfixe mit dem Ordnerkonzept. Wenn Sie nach dem Präfix filtern, werden Objekte mit demselben Präfix in die Metrik-Konfiguration aufgenommen. Weitere Informationen zu Präfixen finden Sie unter [Organisieren von Objekten mit Präfixen](#).
- Tag – Sie können Objekten Markierungen hinzufügen, d. h. Schlüsselwert-Namenspaare. Mit Markierungen können Sie Objekte einfacher finden und organisieren. Markierungen können auch als Filter für Metrik-Konfigurationen verwendet werden. Weitere Informationen über Objekt-Markierungen finden Sie unter [Kategorisieren des Speichers mithilfe von Markierungen](#).
- Zugriffspunkt – S3-Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind und vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3. Wenn Sie einen Zugriffspunkt-Filter erstellen, enthält Amazon S3 Anforderungen an den Zugriffspunkt, den Sie in der Metrikkonfiguration angeben. Weitere Informationen finden Sie unter [Überwachen und Protokollieren von Zugriffspunkten](#).

Note

Wenn Sie eine Metrikkonfiguration erstellen, die nach Access Point filtert, müssen Sie den Access Point Amazon Resource Name (ARN) verwenden, nicht den Access Point-Alias. Stellen Sie sicher, dass Sie den ARN für den Access Point selbst verwenden, nicht den ARN für ein bestimmtes Objekt. Weitere Informationen zum Access Point ARNs finden Sie unter [Verwenden von Zugriffspunkten](#).

Wenn Sie einen Filter angeben, können nur Anforderungen, die für einzelne Objekte ausgeführt werden, mit dem Filter übereinstimmen und in die gemeldeten Metriken aufgenommen werden. Anfragen wie [DeleteObjects](#) und `ListObjects` Anfragen geben keine Metriken für Konfigurationen mit Filtern zurück.

Um ein komplexeres Filtern anzufordern, wählen Sie zwei oder mehr Elemente aus. Nur Objekte, die alle diese Elemente besitzen, werden in die Metrik-Konfiguration aufgenommen. Wenn Sie keine Filter setzen, werden alle Objekte aus dem Bucket in die Metrikenkonfiguration aufgenommen.

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen des Buckets mit den Objekte aus, für die Sie Anforderungsmetriken anfordern möchten.
3. Wählen Sie den Tab Metrics.
4. Wählen Sie unter Bucket metrics (Bucket-Metriken) View additional charts (Zusätzliche Diagramme anzeigen) aus.
5. Wählen Sie den Tab Request metrics (Anforderungsmetriken).
6. Wählen Sie Create Filter (Filter erstellen).
7. Geben Sie im Feld Filter name (Filtername) Ihren Filternamen ein.

Namen dürfen nur Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten.

8. Wählen Sie unter Filterumfang Begrenzen Sie den Bereich dieses Filters mit einem Präfix, Objekt-Tags und einem S3-Zugriffspunkt oder einer Kombination aus allen drei.
9. Wählen Sie unter Filtertyp mindestens einen Filtertyp aus: Präfix, Objekt-Tags, oder Zugriffspunkt.
10. Um einen Präfixfilter zu definieren und den Bereich des Filters auf einen einzelnen Pfad zu beschränken, geben Sie im Feld Präfix ein Präfix ein.
11. Um einen Objekt-Tags-Filter zu definieren, wählen Sie unter Objekt-Tags Hinzufügen eines Tags, und geben Sie dann ein Tag Schlüssel und Wert ein.
12. Um einen Access Point-Filter zu definieren, geben Sie im Feld S3 Access Point den Access Point ein oder wählen Sie Browse S3 ARN, um zum Access Point zu navigieren.

⚠ Important

Sie können keinen Zugriffspunkt-Alias eingeben. Sie müssen den ARN für den Access Point selbst eingeben, nicht den ARN für ein bestimmtes Objekt.

13. Wählen Sie Änderungen speichern.

Amazon S3 erstellt einen Filter, der das Präfix, die Tags oder den Zugriffspunkt verwendet, den Sie angegeben haben.

14. Wählen Sie im Tab Request metrics (Anforderungsmetriken) unter Filters (Filter) den Filter aus, den Sie gerade erstellt haben.

Sie haben nun einen Filter erstellt, der den Bereich der Anforderungsmetriken nach Präfix, Objekt-Tags oder Zugriffspunkt einschränkt. Ungefähr 15 Minuten nach CloudWatch Beginn der Verfolgung dieser Anforderungsmetriken können Sie Diagramme für die Metriken sowohl auf Amazon S3 als auch auf CloudWatch Konsolen sehen. Anforderungsmetriken werden zum CloudWatch Standardtarif abgerechnet. Weitere Informationen finden Sie unter [CloudWatch Amazon-Preise](#).

Sie können Anforderungsmetriken auch auf Bucket-Ebene konfigurieren. Weitere Informationen finden Sie unter [Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket](#).

Verwendung der AWS CLI

1. Installieren und einrichten Sie das AWS CLI. Anweisungen finden Sie unter [Installation, Aktualisierung und Deinstallation von AWS CLI](#) in der AWS Command Line Interface Benutzerleitfaden.
2. Öffnen Sie ein -Terminalfenster.
3. Führen Sie einen der folgenden Befehle aus, um eine Metrik-Konfiguration hinzuzufügen:

Example : So filtern Sie nach Präfix

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --  
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":  
{"Prefix":"prefix1"}} ' 
```

Example : So filtern Sie nach Tags

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"Tag": {"Key": "string", "Value": "string"}} '
```

Example : So filtern Sie nach Zugriffspunkt

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"AccessPointArn": "arn:aws:s3:Region:account-id:accesspoint/access-point-name"} '
```

Example : So filtern Sie nach Präfix, Tags und Zugriffspunkt

```
aws s3api put-bucket-metrics-configuration --endpoint https://
s3.Region.amazonaws.com --bucket DOC-EXAMPLE-BUCKET1 --id metrics-config-id --
metrics-configuration '
{
  "Id": "metrics-config-id",
  "Filter": {
    "And": {
      "Prefix": "string",
      "Tags": [
        {
          "Key": "string",
          "Value": "string"
        }
      ],
      "AccessPointArn": "arn:aws:s3:Region:account-id:accesspoint/access-
point-name"
    }
  }
}'
```

Mit dem REST API

Sie können Metrikkonfigurationen auch programmgesteuert mit dem Amazon S3 hinzufügen. REST API Weitere Informationen zum Hinzufügen und Arbeiten mit Metrikkonfigurationen finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference:

- [PUTKonfiguration von Bucket-Metriken](#)
- [GETKonfiguration von Bucket Metric](#)
- [List Bucket-Metrikkonfiguration](#)
- [DELETEKonfiguration von Bucket Metric](#)

Löschen eines Metrikfilters

Sie können einen CloudWatch Amazon-Anforderungsmetrikfilter löschen, wenn Sie ihn nicht mehr benötigen. Wenn Sie einen Filter löschen, werden Ihnen keine Anforderungsmetriken mehr berechnet, die diesen spezifischen Filter verwenden. Ihnen werden jedoch weiter alle anderen vorhandenen Filterkonfigurationen berechnet.

Wenn Sie einen Filter löschen, können Sie den Filter nicht mehr für Anforderungsmetriken verwenden. Das Löschen eines Filters kann nicht rückgängig gemacht werden.

Informationen zum Erstellen eines Anforderungsmetrikfilters finden Sie in den folgenden Themen:

- [Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket](#)
- [Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert](#)

Verwenden der S3-Konsole

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen Ihres Buckets.
3. Wählen Sie den Tab Metrics.
4. Wählen Sie unter Bucket metrics (Bucket-Metriken) View additional charts (Zusätzliche Diagramme anzeigen) aus.
5. Wählen Sie den Tab Request metrics (Anforderungsmetriken).
6. Wählen Sie Manage filters (Filter verwalten).
7. Wählen Sie Ihren Filter aus.

Important

Das Löschen eines Filters kann nicht rückgängig gemacht werden.

8. Wählen Sie Delete (Löschen).

Amazon S3 löscht Ihren Filter.

Mit dem REST API

Sie können Metrikkonfigurationen auch programmgesteuert mit dem Amazon S3 hinzufügen. REST API Weitere Informationen zum Hinzufügen und Arbeiten mit Metrikkonfigurationen finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference:

- [PUTKonfiguration von Bucket-Metriken](#)
- [GETKonfiguration von Bucket Metric](#)
- [List Bucket-Metrikkonfiguration](#)
- [DELETEKonfiguration von Bucket Metric](#)

Amazon-S3-Ereignis-Benachrichtigungen

Sie können die Amazon-S3-Funktion für Ereignis-Benachrichtigungen verwenden, um Benachrichtigungen zu erhalten, wenn bestimmte Ereignisse in Ihrem S3-Bucket eintreten. Um Benachrichtigungen zu aktivieren, fügen Sie eine Benachrichtigungskonfiguration hinzu, die die Ereignisse angibt, die Amazon S3 veröffentlichen soll. Stellen Sie sicher, dass es auch die Ziele angibt, an die Amazon S3 die Benachrichtigungen senden soll. Sie speichern dieses Konfiguration in der einem Bucket zugeordneten Benachrichtigung-Unterressourcen. Weitere Informationen finden Sie unter [Optionen für die Bucket-Konfiguration](#). Amazon S3 bietet Ihnen eine API Möglichkeit, diese Unterressource zu verwalten.

Important

Amazon-S3-Ereignis-Benachrichtigungen sind so konzipiert, dass sie mindestens einmal zugestellt werden. Ereignis-Benachrichtigungen stellen Ereignisse typischerweise in wenigen Sekunden bereit, manchmal kann dies aber auch eine Minute oder länger dauern.

Übersicht über Amazon-S3-Ereignis-Benachrichtigungen

Zurzeit kann Amazon S3 Benachrichtigungen für die folgenden Ereignisse veröffentlichen:

- Neues Objekt erstellte Ereignisse
- Ereignisse zum Entfernen von Objekten
- Wiederherstellen von Objekt-Ereignissen
- Reduzierte Redundanz: Ereignisse bei Verlust von Speicherobjekten (RRS)
- Replikationsereignisse
- S3-Lebenszyklus-Ablaufereignisse
- S3-Lebenszyklusübergangereignisse
- Aktivieren der automatischen S3-Intelligent-Tiering-Archivierung
- Ereignismarkierungen von Objekten
- Objekt ereignisse ACL PUT

Eine vollständige Beschreibung aller unterstützten Ereignistypen finden Sie unter [Unterstützte Ereignistypen für SQSSNS, und Lambda](#).

Amazon S3 kann Ereignis-Benachrichtigungsmeldungen an die folgenden Ziele senden. Sie geben den Amazon Resource Name (ARN) -Wert dieser Ziele in der Benachrichtigungskonfiguration an.

- Themen zu Amazon Simple Notification Service (AmazonSNS)
- Warteschlangen von Amazon Simple Queue Service (AmazonSQS)
- AWS Lambda function
- Amazon EventBridge

Weitere Informationen finden Sie unter [Unterstützte Ereignisziele](#).

Note

Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter [Amazon aktivieren EventBridge](#).

Warning

Wenn Ihre Benachrichtigung in den selben Bucket schreibt, der die Benachrichtigung auslöst, kann dies zu einer Ausführungsschleife führen. Wenn der Bucket z. B. eine Lambda-Funktion immer dann auslöst, wenn ein Objekt hochgeladen wird, und die Funktion ein Objekt in den Bucket hochlädt, löst sich die Funktion indirekt selbst aus. Um dies zu vermeiden, verwenden Sie zwei Buckets oder konfigurieren Sie den Auslöser so, dass er nur für einen Präfix gilt, der für eingehende Objekte verwendet wird.

Weitere Informationen und ein Beispiel für die Verwendung von Amazon S3 S3-Benachrichtigungen mit AWS Lambda, siehe [Verwenden AWS Lambda mit Amazon S3](#) in der AWS Lambda Leitfaden für Entwickler.

Weitere Informationen zur Anzahl der Konfigurationen für Ereignisbenachrichtigungen, die Sie pro Bucket erstellen können, finden Sie unter [Amazon S3 S3-Servicekontingente](#) unter AWS Allgemeine Referenz.

Weitere Informationen zu Ereignis-Benachrichtigungen finden Sie in den folgenden Abschnitten.

Themen

- [Ereignis-Benachrichtigungstypen und -ziele](#)
- [Verwendung von Amazon SQSSNS, Amazon und Lambda](#)
- [Verwenden EventBridge](#)

Ereignis-Benachrichtigungstypen und -ziele

Amazon S3 unterstützt verschiedene Arten von Ereignis-Benachrichtigungen und Ziele, wo die Benachrichtigungen veröffentlicht werden können. Sie können den Ereignistyp und das Ziel angeben, wenn Sie Ihre Ereignis-Benachrichtigungen konfigurieren. Für jede Ereignisbenachrichtigung kann nur ein Ziel angegeben werden. Amazon-S3-Ereignisbenachrichtigungen senden für jede Benachrichtigungsmeldung einen Ereigniseintrag.

Themen

- [Unterstützte Ereignisziele](#)
- [Unterstützte Ereignistypen für SQSSNS, und Lambda](#)
- [Unterstützte Ereignistypen für Amazon EventBridge](#)

- [Bestellung von Veranstaltungen und doppelte Ereignisse](#)

Unterstützte Ereignisziele

Amazon S3 kann Ereignis-Benachrichtigungsmeldungen an die folgenden Ziele senden.

- Themen zu Amazon Simple Notification Service (AmazonSNS)
- Warteschlangen von Amazon Simple Queue Service (AmazonSQS)
- AWS Lambda
- Amazon EventBridge

Für jede Ereignisbenachrichtigung kann jedoch nur ein Zieltyp angegeben werden.

Note

Sie müssen Amazon S3 S3-Berechtigungen erteilen, um Nachrichten an ein SNS Amazon-Thema oder eine SQS Amazon-Warteschlange zu posten. Sie müssen Amazon S3 außerdem die Erlaubnis erteilen, eine aufzurufen AWS Lambda in Ihrem Namen tätig sein. Anweisungen zum Erteilen dieser Berechtigungen finden Sie unter [Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel](#).

SNSAmazon-Thema

Amazon SNS ist ein flexibler, vollständig verwalteter Push-Messaging-Dienst. Sie können diesen Dienst verwenden, um Nachrichten an mobile Geräte oder verteilte Services zu senden. Mit SNS können Sie eine Nachricht einmal veröffentlichen und sie ein- oder mehrmals zustellen. Derzeit SNS ist Standard nur als Ziel für S3-Ereignisbenachrichtigungen SNS FIFO zulässig, nicht jedoch.

Amazon koordiniert und verwaltet SNS sowohl den Versand als auch die Zustellung von Nachrichten an abonnierte Endpunkte oder Kunden. Sie können die SNS Amazon-Konsole verwenden, um ein SNS Amazon-Thema zu erstellen, an das Ihre Benachrichtigungen gesendet werden können.

Das Thema muss dasselbe sein AWS-Region als Ihr Amazon S3 S3-Bucket. Anweisungen zum Erstellen eines SNS Amazon-Themas finden Sie unter [Erste Schritte mit Amazon SNS im Amazon Simple Notification Service Developer Guide](#) und im [Amazon SNS FAQ](#).

Bevor Sie das von Ihnen erstellte SNS Amazon-Thema als Ziel für Ereignisbenachrichtigungen verwenden können, benötigen Sie Folgendes:

- Der Amazon-Ressourcenname (ARN) für das SNS Amazon-Thema
- Ein gültiges SNS Amazon-Themenabonnement. Damit werden Themenabonnenten benachrichtigt, wenn eine Nachricht zu Ihrem SNS Amazon-Thema veröffentlicht wird.

SQSAmazon-Warteschlange

Amazon SQS bietet zuverlässige und skalierbare gehostete Warteschlangen zum Speichern von Nachrichten, während sie zwischen Computern übertragen werden. Sie können Amazon verwenden SQS, um beliebige Datenmengen zu übertragen, ohne dass andere Dienste immer verfügbar sein müssen. Sie können die SQS Amazon-Konsole verwenden, um eine SQS Amazon-Warteschlange zu erstellen, an die Ihre Benachrichtigungen gesendet werden können.

Die SQS Amazon-Warteschlange muss sich in derselben befinden AWS-Region als Ihr Amazon S3 S3-Bucket. Anweisungen zum Erstellen einer SQS Amazon-Warteschlange finden Sie unter [Was ist Amazon Simple Queue Service](#) und [Erste Schritte mit Amazon SQS im Amazon Simple Queue Service Developer Guide](#).

Bevor Sie die SQS Amazon-Warteschlange als Ziel für Ereignisbenachrichtigungen verwenden können, benötigen Sie Folgendes:

- Der Amazon-Ressourcenname (ARN) für die SQS Amazon-Warteschlange

Note

Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter [Amazon aktivieren EventBridge](#).

Lambda-Funktion

Sie können Folgendes verwenden ... AWS Lambda um andere zu erweitern AWS Dienste mit benutzerdefinierter Logik, oder erstellen Sie Ihr eigenes Backend, das unter funktioniert

AWS Skalierung, Leistung und Sicherheit. Mit Lambda können Sie diskrete ereignisabhängige Anwendungen erstellen, die nur ausgeführt werden, wenn sie benötigt werden. Sie können damit auch diese Anwendungen automatisch von ein paar Anfragen pro Tag auf Tausende pro Sekunde skalieren.

Lambda kann benutzerdefinierten Code in Reaktion auf Amazon-S3-Bucket-Ereignisse ausführen. Sie laden Ihren benutzerdefinierten Code auf Lambda hoch und erstellen eine sogenannte Lambda-Funktion. Wenn Amazon S3 ein Ereignis eines bestimmten Typs erkennt, kann es das Ereignis veröffentlichen AWS Lambda und rufe deine Funktion in Lambda auf. Lambda führt als Antwort Ihre Funktion aus. Ein Ereignistyp, den er möglicherweise erkennt, ist beispielsweise ein objekterstelltes Ereignis.

Sie können das AWS Lambda Konsole zum Erstellen einer Lambda-Funktion, die die AWS Infrastruktur, um den Code in Ihrem Namen auszuführen. Die Lambda-Funktion muss sich in derselben Region wie Ihr S3-Bucket befinden. Sie müssen auch den Namen oder den ARN einer Lambda-Funktion haben, um die Lambda-Funktion als Ziel für Ereignisbenachrichtigungen einzurichten.

Warning

Wenn Ihre Benachrichtigung in den selben Bucket schreibt, der die Benachrichtigung auslöst, kann dies zu einer Ausführungsschleife führen. Wenn der Bucket z. B. eine Lambda-Funktion immer dann auslöst, wenn ein Objekt hochgeladen wird, und die Funktion ein Objekt in den Bucket hochlädt, löst sich die Funktion indirekt selbst aus. Um dies zu vermeiden, verwenden Sie zwei Buckets oder konfigurieren Sie den Auslöser so, dass er nur für einen Präfix gilt, der für eingehende Objekte verwendet wird.

Weitere Informationen und ein Beispiel für die Verwendung von Amazon S3 S3-Benachrichtigungen mit AWS Lambda, siehe [Verwenden AWS Lambda mit Amazon S3](#) in der AWS Lambda Leitfaden für Entwickler.

Amazon EventBridge

Amazon EventBridge ist ein serverloser Event-Bus, der Ereignisse empfängt von AWS Dienstleistungen. Sie können Regeln einrichten, um Ereignisse zuzuordnen und sie an Ziele weiterzuleiten, z. B. AWS Dienst oder ein HTTP Endpunkt. Weitere Informationen finden Sie unter [Was steht EventBridge](#) im EventBridge Amazon-Benutzerhandbuch.

Im Gegensatz zu anderen Zielen können Sie EventBridge für einen Bucket Ereignisse, an die gesendet werden sollen, entweder aktivieren oder deaktivieren. Wenn Sie die Zustellung aktivieren, werden alle Ereignisse an gesendet EventBridge. Darüber hinaus können Sie EventBridge Regeln verwenden, um Ereignisse an weitere Ziele weiterzuleiten.

Unterstützte Ereignistypen für SQSSNS, und Lambda

Amazon S3 kann Ereignisse der folgenden Typen veröffentlichen. Sie spezifizieren diese Ereignistypen in der Benachrichtigungskonfiguration.

Ereignistypen	Beschreibung
s3:TestEvent	<p>Wenn eine Benachrichtigung aktiviert ist, veröffentlicht Amazon S3 eine Testbenachrichtigung. Damit soll sichergestellt werden, dass das Thema vorhanden ist und dass der Bucket-Eigentümer über die Berechtigung verfügt, das angegebene Thema zu veröffentlichen.</p> <p>Wenn das Aktivieren der Benachrichtigung fehlschlägt, erhalten Sie keine Testbenachrichtigung.</p>
s3:ObjectCreated:* s3:ObjectCreated:Put s3:ObjectCreated:Post s3:ObjectCreated:Copy s3:ObjectCreated:CompleteMultipartUpload	<p>Amazon S3 API S3-Operationen wie PUTPOST, und COPY können ein Objekt erstellen. Mit diesen Ereignistypen können Sie Benachrichtigungen aktivieren, wenn ein Objekt mithilfe eines bestimmten API Vorgangs erstellt wird. Alternativ können Sie den s3:ObjectCreated:* Ereignistyp verwenden, um Benachrichtigungen anzufordern, unabhängig davon API, mit welchem Ereignistyp ein Objekt erstellt wurde.</p> <p>s3:ObjectCreated:CompleteMultipartUpload schließt Objekte ein, die erstellt wurden mit UploadPartCopy für Kopiervorgänge.</p>
s3:ObjectRemoved:* s3:ObjectRemoved>Delete	<p>Mithilfe der ObjectRemoved Ereignistypen können Sie Benachrichtigungen aktivieren, wenn ein Objekt oder ein Stapel von Objekten aus einem Bucket entfernt wird.</p>

Ereignistypen	Beschreibung
<p>s3:ObjectRemoved:DeleteMarkerCreated</p>	<p>Sie können eine Benachrichtigung anfordern, wenn ein Objekt gelöscht wird, oder wenn ein versionsfähiges Objekt permanent gelöscht wird. Dazu wird der Ereignistyp <code>s3:ObjectRemoved:Delete</code> verwendet. Alternativ können Sie eine Benachrichtigung anfordern, wenn eine Löschmarkierung für ein versioniertes Objekt mit <code>s3:ObjectRemoved:DeleteMarkerCreated</code> erstellt wird. Eine entsprechende Anleitung zum Löschen von versionierten Objekten finden Sie unter Löschen von Objekten aus einem versioning-fähigen Bucket. Sie können auch eine Wildcard <code>s3:ObjectRemoved:*</code> verwenden, um bei jedem Löschen eines Objekts eine Benachrichtigung anzufordern.</p> <p>Diese Ereignisbenachrichtigungen warnen Sie nicht vor automatischen Löschungen aufgrund von Lebenszykluskonfigurationen oder fehlgeschlagenen Operationen.</p>
<p>s3:ObjectRestore:* s3:ObjectRestore:Post s3:ObjectRestore:Completed s3:ObjectRestore:Delete</p>	<p>Mithilfe der <code>ObjectRestore</code> Ereignistypen können Sie Benachrichtigungen über die Initiierung und den Abschluss von Ereignissen erhalten, wenn Objekte aus der Speicherklasse S3 Glacier Flexible Retrieval, der Speicherklasse S3 Glacier Deep Archive, der Stufe S3 Intelligent-Tiering Archive Access und der Stufe S3 Intelligent-Tiering Deep Archive Access wiederhergestellt werden. Sie können auch Benachrichtigungen erhalten, wenn die wiederhergestellte Kopie eines Objekts abläuft.</p> <p>Der <code>s3:ObjectRestore:Post</code> -Ereignistyp informiert Sie über die Initiierung von Objektwiederherstellungen. Der <code>s3:ObjectRestore:Completed</code> -Ereignistyp informiert Sie über den Abschluss von Wiederherstellungen. Der <code>s3:ObjectRestore:Delete</code> -Ereignistyp benachrichtigt Sie, wenn die temporäre Kopie eines wiederhergestellten Objekts abläuft.</p>

Ereignistypen	Beschreibung
s3:ReducedRedundancyLostObject	Sie erhalten dieses Benachrichtigungsereignis, wenn Amazon S3 feststellt, dass ein Objekt der RRS Speicherklasse verloren gegangen ist.

Ereignistypen	Beschreibung
<p>s3:Replication:*</p> <p>s3:Replication:OperationFailedReplication</p> <p>s3:Replication:OperationMissedThreshold</p> <p>s3:Replication:OperationReplicatedAfterThreshold</p> <p>s3:Replication:OperationNotTracked</p>	<p>Mithilfe der Replication Ereignistypen können Sie Benachrichtigungen für Replikationskonfigurationen erhalten, für die S3-Replikationsmetriken oder S3 Replication Time Control (S3RTC) aktiviert sind. Sie können den minute-by-minute Fortschritt von Replikationsereignissen überwachen, indem Sie ausstehende Byte, ausstehende Vorgänge und die Replikationslatenz verfolgen. Informationen zu Replikationsmetriken finden Sie unter Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status.</p> <ul style="list-style-type: none"> • Der s3:Replication:OperationFailedReplication -Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation berechtigt war, nicht repliziert werden konnte. • Der s3:Replication:OperationMissedThreshold Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, den 15-minütigen Schwellenwert für die Replikation RTC überschreitet. • Der s3:Replication:OperationReplicatedAfterThreshold Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, nach dem RTC 15-Minuten-Schwellenwert repliziert wird. • Der s3:Replication:OperationNotTracked Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Live-Replikation in Frage kam (entweder regionsübergreifende Replikation [SRR] oder regionsübergreifende Replikation [CRR]), nicht mehr anhand von Replikationsmetriken verfolgt wird.

Ereignistypen	Beschreibung
<p><code>s3:LifecycleExpiration:*</code></p> <p><code>s3:LifecycleExpiration:Delete</code></p> <p><code>s3:LifecycleExpiration:DeleteMarkerCreated</code></p>	<p>Mithilfe der <code>LifecycleExpiration</code> Ereignistypen können Sie eine Benachrichtigung erhalten, wenn Amazon S3 ein Objekt auf der Grundlage Ihrer S3-Lifecycle-Konfiguration löscht.</p> <p>Der Ereignistyp <code>s3:LifecycleExpiration:Delete</code> benachrichtigt Sie, wenn ein Objekt in einem unversionierten Bucket gelöscht wird. Es benachrichtigt Sie auch, wenn eine Objektversion durch eine S3-Lebenszyklus-Konfiguration dauerhaft gelöscht wird. Der Ereignistyp <code>s3:LifecycleExpiration:DeleteMarkerCreated</code> benachrichtigt Sie, wenn S3 Lebenszyklus eine Löschmarke erstellt, wenn eine aktuelle Version eines Objekts im versionierten Bucket gelöscht wird.</p>
<p><code>s3:LifecycleTransition</code></p>	<p>Sie erhalten dieses Benachrichtigungsereignis, wenn ein Objekt von einer S3-Lebenszykluskonfiguration in eine andere Amazon-S3-Speicherklasse überführt wird.</p>
<p><code>s3:IntelligentTiering</code></p>	<p>Sie erhalten dieses Benachrichtigungsereignis, wenn ein Objekt innerhalb der Speicherklasse S3 Intelligent-Tiering in die Stufe Archive Access oder Deep Archive Access verschoben wird.</p>
<p><code>s3:ObjectTagging:*</code></p> <p><code>s3:ObjectTagging:Put</code></p> <p><code>s3:ObjectTagging>Delete</code></p>	<p>Mithilfe der <code>ObjectTagging</code> Ereignistypen können Sie Benachrichtigungen aktivieren, wenn ein Objekt-Tag hinzugefügt oder aus einem Objekt gelöscht wird.</p> <p>Der <code>s3:ObjectTagging:Put</code> Ereignistyp benachrichtigt Sie, wenn sich ein Tag PUT auf einem Objekt befindet oder wenn ein vorhandenes Tag aktualisiert wird. Der <code>s3:ObjectTagging>Delete</code> -Ereignistyp benachrichtigt Sie, wenn ein Tag aus einem Objekt entfernt wird.</p>

Ereignistypen	Beschreibung
s3:ObjectAc1:Put	Sie erhalten dieses Benachrichtigungsereignis, wenn ACL sich ein Objekt PUT auf einem Objekt befindet oder wenn ein vorhandenes geändert ACL wird. Ein Ereignis wird nicht generiert, wenn eine Anfrage zu keiner Änderung an einem Objekt führtACL.

Unterstützte Ereignistypen für Amazon EventBridge

Eine Liste der Ereignistypen, die Amazon S3 an Amazon sendet EventBridge, finden Sie unter [Verwenden EventBridge](#).

Bestellung von Veranstaltungen und doppelte Ereignisse

Amazon S3 Event Notifications ist so konzipiert, dass Benachrichtigungen mindestens einmal zugestellt werden, aber es kann nicht garantiert werden, dass sie in derselben Reihenfolge ankommen, in der die Ereignisse eingetreten sind. In seltenen Fällen kann der Wiederholungsmechanismus von Amazon S3 zu doppelten S3-Ereignisbenachrichtigungen für dasselbe Objekt ereignis führen. Weitere Informationen zum Umgang mit doppelten oder nicht in der richtigen Reihenfolge befindlichen Ereignissen finden Sie unter [Verwaltung der Bestellung von Veranstaltungen und doppelten Ereignissen mit Amazon S3 S3-Ereignisbenachrichtigungen](#) auf der AWS Blog zum Thema Speicher.

Verwendung von Amazon SQSSNS, Amazon und Lambda

Die Aktivierung von Benachrichtigungen erfolgt auf Bucket-Ebene. Sie speichern Informationen zur Benachrichtigungskonfiguration in der Benachrichtigungen-Unterressourcen die einem Bucket zugeordnet sind. Nachdem Sie die Bucket-Benachrichtigungskonfiguration erstellt oder geändert haben, dauert es normalerweise etwa fünf Minuten, bis die Änderungen wirksam werden. Ein s3:TestEvent tritt auf, wenn die Benachrichtigung zum ersten Mal aktiviert wird. Sie können jede der folgenden Methoden verwenden, um die Benachrichtigungskonfiguration zu verwalten:

- Verwenden der Amazon-S3-Konsole – Die Benutzeroberfläche der Konsole ermöglicht Ihnen, eine Benachrichtigungskonfiguration für einen Bucket einzurichten, ohne Code schreiben zu müssen. Weitere Informationen finden Sie unter [Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole](#).

- Programmgesteuert mit dem AWS SDKs— Intern SDKs rufen sowohl die Konsole als auch Amazon S3 REST API auf, um die Unterressourcen für Benachrichtigungen zu verwalten, die dem Bucket zugeordnet sind. Hier finden Sie Beispiele für Benachrichtigungskonfigurationen, die Folgendes verwenden AWS SDK, siehe [Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen \(SNSThema oder SQS Warteschlange\)](#).

Note

Sie können die Amazon S3 REST API S3-Anrufe auch direkt von Ihrem Code aus tätigen. Dies kann jedoch umständlich sein, da Sie dazu Code schreiben müssen, um Ihre Anforderungen zu authentifizieren.

Unabhängig von der Methode, die Sie verwenden, speichert Amazon S3 die Benachrichtigungskonfiguration wie XML in der Benachrichtigungs-Subressource, die einem Bucket zugeordnet ist. Weitere Informationen zu Bucket-Subressourcen finden Sie unter [Optionen für die Bucket-Konfiguration](#).

Note

Wenn Sie aufgrund gelöschter Ziele mehrere fehlgeschlagene Ereignisbenachrichtigungen haben, erhalten Sie möglicherweise die Meldung Die folgenden Zielkonfigurationen konnten nicht überprüft werden, wenn Sie versuchen, sie zu löschen. Sie können dieses Problem in der S3-Konsole beheben, indem Sie alle fehlgeschlagenen Benachrichtigungen gleichzeitig löschen.

Themen

- [Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel](#)
- [Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole](#)
- [Programmgesteuerte Konfiguration von Ereignis-Benachrichtigungen](#)
- [Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen \(SNSThema oder SQS Warteschlange\)](#)
- [Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung](#)
- [Struktur von Ereignismeldungen](#)

Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel

Sie müssen dem Amazon S3 S3-Principal die erforderlichen Berechtigungen zum Aufrufen der relevanten Funktionen API zur Veröffentlichung von Nachrichten in einem SNS Thema, einer SQS Warteschlange oder einer Lambda-Funktion gewähren. So kann Amazon S3 Ereignisbenachrichtigungsmeldungen an einem Ziel veröffentlichen.

Informationen zur Fehlerbehebung beim Veröffentlichen von Ereignisbenachrichtigungen in einem Ziel finden Sie unter [Fehlerbehebung bei der Veröffentlichung von Amazon-S3-Ereignisbenachrichtigungen in einem Thema von Amazon Simple Notification Service](#).

Themen

- [Erteilen von Berechtigungen zum Aufrufen eines AWS Lambda function](#)
- [Erteilen von Berechtigungen zum Veröffentlichen von Nachrichten zu einem SNS Thema oder einer SQS Warteschlange](#)

Erteilen von Berechtigungen zum Aufrufen eines AWS Lambda function

Amazon S3 veröffentlicht Ereignisnachrichten an AWS Lambda indem Sie eine Lambda-Funktion aufrufen und die Ereignisnachricht als Argument bereitstellen.

Wenn Sie die Amazon-S3-Konsole verwenden, um Ereignisbenachrichtigungen in einem Amazon-S3-Bucket für eine Lambda-Funktion zu konfigurieren, richtet die Konsole die erforderlichen Berechtigungen für die Lambda-Funktion ein. Dies ist so, dass Amazon S3 über Berechtigungen verfügt, die Funktion aus dem Bucket aufzurufen. Weitere Informationen finden Sie unter [Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole](#).

Sie können Amazon S3 S3-Berechtigungen auch erteilen von AWS Lambda um Ihre Lambda-Funktion aufzurufen. Weitere Informationen finden Sie unter [Tutorial: Verwenden AWS Lambda mit Amazon S3](#) in der AWS Lambda Leitfaden für Entwickler.

Erteilen von Berechtigungen zum Veröffentlichen von Nachrichten zu einem SNS Thema oder einer SQS Warteschlange

Um Amazon S3 S3-Berechtigungen zur Veröffentlichung von Nachrichten im SNS Thema oder in der SQS Warteschlange zu erteilen, fügen Sie eine AWS Identity and Access Management (IAM) Richtlinie für das SNS Zielthema oder die SQS Zielwarteschlange.

Ein Beispiel dafür, wie Sie eine Richtlinie an ein SNS Thema oder eine SQS Warteschlange anhängen, finden Sie unter [Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen \(SNS Thema oder SQS Warteschlange\)](#). Weitere Informationen über Berechtigungen finden Sie in den folgenden Themen:

- [Beispielfälle für Amazon SNS Access Control](#) im Amazon Simple Notification Service Developer Guide
- [Identitäts- und Zugriffsmanagement in Amazon SQS](#) im Amazon Simple Queue Service Developer Guide

IAM Richtlinie für ein SNS Zielthema

Das Folgende ist ein Beispiel für ein AWS Identity and Access Management (IAM) Richtlinie, die Sie an das SNS Zielthema anhängen. Anweisungen dazu, wie Sie mithilfe dieser Richtlinie ein SNS Amazon-Zielthema für Ereignisbenachrichtigungen einrichten können, finden Sie unter [Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen \(SNS Thema oder SQS Warteschlange\)](#).

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "Example SNS topic policy",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "SNS-topic-ARN",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:bucket-name"
        },
        "StringEquals": {
          "aws:SourceAccount": "bucket-owner-account-id"
        }
      }
    }
  ]
}
```

```
]
}
```

IAM Richtlinie für eine SQS Zielwarteschlange

Im Folgenden finden Sie ein Beispiel IAM für eine Richtlinie, die Sie an die SQS Zielwarteschlange anhängen. Anweisungen dazu, wie Sie diese Richtlinie verwenden können, um eine SQS Amazon-Zielwarteschlange für Ereignisbenachrichtigungen einzurichten, finden Sie unter [Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen \(SNS Thema oder SQS Warteschlange\)](#).

Um diese Richtlinie verwenden zu können, müssen Sie die SQS Amazon-Warteschlange ARN, den Bucket-Namen und den Bucket-Besitzer aktualisieren AWS-Konto ID.

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SQS:SendMessage"
      ],
      "Resource": "arn:aws:sqs:Region:account-id:queue-name",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
        },
        "StringEquals": {
          "aws:SourceAccount": "bucket-owner-account-id"
        }
      }
    }
  ]
}
```

Sowohl für die Amazon SNS - als auch für SQS IAM die Amazon-Richtlinien können Sie anstelle der StringLike Bedingung die ArnLike Bedingung in der Richtlinie angeben.

Wenn diese ArnLike Option verwendet wird, müssen die Partition, der Service, die Account-ID, der Ressourcentyp und der Teil der Ressourcen-ID exakt mit dem im ARN Anforderungskontext übereinstimmen. ARN Nur die Region und der Ressourcenpfad lassen einen teilweisen Abgleich zu.

Wenn anstelle von verwendet StringLike wird ArnLike, ignoriert der Abgleich die ARN Struktur und ermöglicht einen teilweisen Abgleich, unabhängig davon, welcher Teil mit einem Platzhalter versehen wurde. Weitere Informationen finden Sie unter [IAMJSONRichtlinienelemente](#) im IAM Benutzerhandbuch.

```
"Condition": {
  "StringLike": { "aws:SourceArn": "arn:aws:s3:*:*:bucket-name" }
}
```

AWS KMS wichtige Richtlinie

Wenn die SQS Warteschlange oder die SNS Themen mit einem verschlüsselt sind AWS Key Management Service (AWS KMS) vom Kunden verwalteter Schlüssel, müssen Sie dem Amazon S3 S3-Service Principal die Erlaubnis erteilen, mit den verschlüsselten Themen oder der Warteschlange zu arbeiten. Um dem Amazon-S3-Service-Prinzipal die Berechtigung zu erteilen, fügen Sie der Schlüsselrichtlinie für den vom Kunden verwalteten Schlüssel die folgende Anweisung hinzu.

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen zur AWS KMS wichtige Richtlinien finden Sie unter [Verwenden wichtiger Richtlinien in AWS KMS](#) in der AWS Key Management Service Leitfaden für Entwickler.

Weitere Informationen zur Verwendung der serverseitigen Verschlüsselung mit AWS KMS Informationen zu Amazon SQS und Amazon SNS finden Sie im Folgenden:

- [Schlüsselverwaltung](#) im Amazon Simple Notification Service-Entwicklerhandbuch.
- [Schlüsselverwaltung](#) im Amazon Simple Queue Service-Entwicklerhandbuch.
- [Verschlüsseln von Nachrichten, die auf Amazon SNS veröffentlicht wurden, mit AWS KMS](#) in der AWS Blog berechnen.

Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole

Sie können bestimmte Amazon-S3-Bucket-Ereignisse aktivieren, um eine Benachrichtigungsmeldung an ein Ziel zu senden, wenn das Ereignis auftritt. In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole Ereignis-Benachrichtigungen aktivieren können. Informationen zur Verwendung von Ereignisbenachrichtigungen mit dem AWS SDKs und der Amazon S3 REST APIs, siehe [Programmgesteuerte Konfiguration von Ereignis-Benachrichtigungen](#).

Voraussetzungen: Bevor Sie Ereignis-Benachrichtigungen für Ihren Bucket aktivieren können, müssen Sie einen der Zieltypen einrichten und dann die Berechtigungen konfigurieren. Weitere Informationen erhalten Sie unter [Unterstützte Ereignisziele](#) und [Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel](#).

Note

Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter [Amazon aktivieren EventBridge](#).

Themen

- [Amazon- SNSSQS, Amazon- oder Lambda-Benachrichtigungen über die Amazon S3 S3-Konsole aktivieren](#)

Amazon- SNS, Amazon- oder Lambda-Benachrichtigungen über die Amazon S3 S3-Konsole aktivieren

Ereignis-Benachrichtigungen für einen S3-Bucket aktivieren und konfigurieren

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Ereignisse aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Navigieren Sie zum Abschnitt Event Notifications (Ereignis-Benachrichtigungen) und wählen Sie Create event notification (Ereignis-Benachrichtigung erstellen)
5. Geben Sie im Abschnitt General configuration (Allgemeine Konfiguration) den beschreibenden Ereignisnamen für Ihre Ereignis-Benachrichtigung an. Optional können Sie auch ein Präfix und ein Suffix angeben, um die Benachrichtigungen auf Objekte mit Schlüsseln zu beschränken, die mit den angegebenen Zeichen enden.

- a. Geben Sie eine Beschreibung für den Namen der Veranstaltung ein.

Wenn Sie keinen Namen eingeben, wird eine global eindeutige Kennung (GUID) generiert und für den Namen verwendet.

- b. (Optional) Um Ereignisbenachrichtigungen nach Präfix zu filtern, geben Sie ein Präfix ein.

Sie können beispielsweise einen Präfix-Filter einrichten, damit Sie nur Benachrichtigungen erhalten, wenn Dateien zu einem bestimmten Ordner (z. B.) hinzugefügt werden, `images/`).

- c. (Optional) Um Ereignisbenachrichtigungen nach Suffix zu filtern, geben Sie ein Suffix ein.

Weitere Informationen finden Sie unter [Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung](#).

6. Wählen Sie im Abschnitt Event types (Ereignistypen) einen oder mehrere Ereignistypen aus, für die Sie Benachrichtigungen erhalten möchten.

Eine Liste der verschiedenen Ereignistypen finden Sie unter [Unterstützte Ereignistypen für SQS, SNS, und Lambda](#).

7. Wählen Sie im Abschnitt Destination (Ziel) das Ziel für Ereignis-Benachrichtigungen aus.

Note

Bevor Sie Ereignisbenachrichtigungen veröffentlichen können, müssen Sie dem Amazon S3-Principal die erforderlichen Berechtigungen erteilen, um die entsprechenden Personen aufzurufenAPI. Auf diese Weise können Benachrichtigungen für eine Lambda-Funktion, ein SNS Thema oder eine SQS Warteschlange veröffentlicht werden.

- a. Wählen Sie den Zieltyp aus: Lambda-Funktion, SNSThema oder SQSWarteschlange.
- b. Nachdem Sie Ihren Zieltyp ausgewählt haben, wählen Sie eine Funktion, ein Thema oder eine Warteschlange aus der Liste aus.
- c. Oder, wenn Sie lieber einen Amazon-Ressourcennamen (ARN) angeben möchten, wählen Sie Enter ARN und geben Sie den einARN.

Weitere Informationen finden Sie unter [Unterstützte Ereignisziele](#).

8. Wählen Sie Save changes (Änderungen speichern) und Amazon S3 sendet eine Testnachricht an das Ziel für Ereignis-Benachrichtigungen.

Programmgesteuerte Konfiguration von Ereignis-Benachrichtigungen

Standardmäßig sind für keinen Ereignistyp Benachrichtigungen aktiviert. Aus diesem Grund speichert die Benachrichtigung-Unterressourcen anfänglich eine leere Konfiguration.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</NotificationConfiguration>
```

Um Benachrichtigungen für Ereignisse bestimmter Typen zu aktivieren, ersetzen Sie die durch die XML entsprechende Konfiguration, die die Ereignistypen, die Amazon S3 veröffentlichen soll, und das Ziel, an dem die Ereignisse veröffentlicht werden sollen, identifiziert. Für jedes Ziel fügen Sie eine entsprechende XML Konfiguration hinzu.

Um Ereignisnachrichten in einer SQS Warteschlange zu veröffentlichen

Um eine SQS Warteschlange als Benachrichtigungsziel für einen oder mehrere Ereignistypen festzulegen, fügen Sie die hinzuQueueConfiguration.


```

<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </QueueConfiguration>
  ...
</NotificationConfiguration>

```

Um Ereignisnachrichten zu einem SNS Thema zu veröffentlichen

Um ein SNS Thema als Benachrichtigungsziel für bestimmte Ereignistypen festzulegen, fügen Sie den hinzu `TopicConfiguration`.

```

<NotificationConfiguration>
  <TopicConfiguration>
    <Id>optional-id-string</Id>
    <Topic>sns-topic-arn</Topic>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </TopicConfiguration>
  ...
</NotificationConfiguration>

```

Um das aufzurufen AWS Lambda Funktion und Bereitstellung einer Ereignisnachricht als Argument

Um eine Lambda-Funktion als Benachrichtigungsziel für bestimmte Ereignistypen festzulegen, fügen Sie `CloudFunctionConfiguration` hinzu.

```

<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>optional-id-string</Id>
    <CloudFunction>cloud-function-arn</CloudFunction>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </CloudFunctionConfiguration>
  ...
</NotificationConfiguration>

```

So entfernen Sie alle in einem Bucket konfigurierten Benachrichtigungen

Um alle für einen Bucket konfigurierten Benachrichtigungen zu entfernen, speichern Sie ein leeres `<NotificationConfiguration/>`-Element in der notification-Subressource.

Wenn Amazon S3 ein Ereignis des spezifischen Typs erkennt, veröffentlicht es eine Meldung mit der Ereignisinformation. Weitere Informationen finden Sie unter [Struktur von Ereignismeldungen](#).

Weitere Informationen zur Konfiguration von Ereignis-Benachrichtigungen finden Sie in den folgenden Themen:

- [Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen \(SNSThema oder SQS Warteschlange\)](#).
- [Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung](#)

Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange)

Sie können Amazon S3 S3-Benachrichtigungen mit Amazon Simple Notification Service (AmazonSNS) oder Amazon Simple Queue Service (AmazonSQS) erhalten. In dieser exemplarischen Vorgehensweise fügen Sie Ihrem Bucket mithilfe eines SNS Amazon-Themas und einer SQS Amazon-Warteschlange eine Benachrichtigungskonfiguration hinzu.

Note

Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter [Amazon aktivieren EventBridge](#).

Themen

- [Walkthrough-Übersicht](#)
- [Schritt 1: Erstellen Sie eine SQS Amazon-Warteschlange](#)
- [Schritt 2: Erstellen Sie ein SNS Amazon-Thema](#)
- [Schritt 3: Hinzufügen einer Benachrichtigungskonfiguration zu Ihrem Bucket](#)

- [Schritt 4: Testen der Einrichtung](#)

Walkthrough-Übersicht

Dieser Walkthrough hilft Ihnen bei folgenden Aufgaben:

- Veröffentlichen Sie Ereignisse dieses `s3:ObjectCreated:*` Typs in einer SQS Amazon-Warteschlange.
- Veröffentlichen Sie Ereignisse dieses `s3:ReducedRedundancyLostObject` Typs zu einem SNS Amazon-Thema.

Weitere Informationen zur Benachrichtigungskonfiguration finden Sie unter [Verwendung von Amazon SQSSNS, Amazon und Lambda](#)

Alle diese Schritte können Sie auf der Konsole erledigen, ohne Code schreiben zu müssen. Darüber hinaus Codebeispiele mit AWS SDKs für Java und .NET werden auch bereitgestellt, um Ihnen beim programmgesteuerten Hinzufügen von Benachrichtigungskonfigurationen zu helfen.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

1. Erstellen Sie eine SQS Amazon-Warteschlange.

Erstellen Sie mit der SQS Amazon-Konsole eine SQS Warteschlange. Sie können auf alle Meldungen zugreifen, die Amazon S3 programmgesteuert an die Warteschlange sendet. Für diese schrittweise Anleitung überprüfen Sie die Benachrichtigungsmittelungen in der Konsole.

Sie ordnen der Warteschlange eine Zugriffsrichtlinie zu, um Amazon S3 die Berechtigung zu erteilen, Meldungen zu veröffentlichen.

2. Erstellen Sie ein SNS Amazon-Thema.

Erstellen Sie mit der SNS Amazon-Konsole ein SNS Thema und abonnieren Sie das Thema. Auf diese Weise werden alle darin veröffentlichten Ereignisse an Sie weitergegeben. Sie geben als Kommunikationsprotokoll E-Mail an. Nachdem Sie ein Thema erstellt haben, SNS sendet Amazon eine E-Mail. Sie verwenden den Link in der E-Mail, um das Abonnement des Themas zu bestätigen.

Sie ordnen dem Thema eine Zugriffsrichtlinie zu, um Amazon S3 die Berechtigung zu erteilen, Meldungen zu veröffentlichen.

3. Fügen Sie einem Bucket eine Benachrichtigungskonfiguration hinzu.

Schritt 1: Erstellen Sie eine SQS Amazon-Warteschlange

Folgen Sie den Schritten, um eine Amazon Simple Queue Service (AmazonSQS) -Warteschlange zu erstellen und zu abonnieren.

1. Erstellen Sie mit der SQS Amazon-Konsole eine Warteschlange. Anweisungen finden Sie unter [Erste Schritte mit Amazon SQS im Amazon Simple Queue Service Developer Guide](#).
2. Ersetzen Sie die der Warteschlange zugeordnete Zugriffsrichtlinie durch die folgende Richtlinie.
 - a. Wählen Sie in der SQS Amazon-Konsole in der Warteschlangenliste den Namen der Warteschlange aus.
 - b. Wählen Sie im Tab Zugriffsrichtlinie die Option Bearbeiten aus.
 - c. Ersetzen Sie die Zugriffsrichtlinie, die der Warteschlange angefügt ist. Geben Sie darin Ihren Amazon-Namen SQSARN, den Quell-Bucket-Namen und die Konto-ID des Bucket-Besitzers an.

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SQS:SendMessage"
      ],
      "Resource": "SQS-queue-ARN",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
        },
        "StringEquals": {
          "aws:SourceAccount": "bucket-owner-account-id"
        }
      }
    }
  ]
}
```

- d. Wählen Sie Save (Speichern) aus.
3. (Optional) Wenn für die SQS Amazon-Warteschlange oder das SNS Amazon-Thema serverseitige Verschlüsselung aktiviert ist mit AWS Key Management Service (AWS KMS), fügen Sie dem zugehörigen, vom Kunden verwalteten Schlüssel für symmetrische Verschlüsselung die folgende Richtlinie hinzu.

Sie müssen die Richtlinie einem vom Kunden verwalteten Schlüssel hinzufügen, da Sie die Richtlinie nicht ändern können AWS verwalteter Schlüssel für Amazon SQS oder AmazonSNS.

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen zur Verwendung von SSE für Amazon SQS und Amazon SNS mit AWS KMS, siehe Folgendes:

- [Schlüsselverwaltung](#) im Amazon Simple Notification Service-Entwicklerhandbuch.
 - [Schlüsselverwaltung](#) im Amazon Simple Queue Service-Entwicklerhandbuch.
4. Beachten Sie die WarteschlangeARN.

Die SQS Warteschlange, die Sie erstellt haben, ist eine weitere Ressource in Ihrem AWS-Konto. Es hat einen eindeutigen Amazon-Ressourcennamen (ARN). Das benötigen Sie ARN im nächsten Schritt. ARN hat das folgende Format:

```
arn:aws:sqs:aws-region:account-id:queue-name
```

Schritt 2: Erstellen Sie ein SNS Amazon-Thema

Folgen Sie den Schritten, um ein SNS Amazon-Thema zu erstellen und zu abonnieren.

1. Erstellen Sie mit der SNS Amazon-Konsole ein Thema. Anweisungen finden Sie unter [Erstellen eines SNS Amazon-Themas](#) im Amazon Simple Notification Service Developer Guide.
2. Abonnieren Sie das Thema. Für diese Übung geben Sie email als Kommunikationsprotokoll an. Eine Anleitung finden Sie unter [SNS Amazon-Thema abonnieren im Amazon Simple Notification Service Developer Guide](#).

Sie erhalten eine E-Mail, in der Sie aufgefordert werden, das Abonnement des Themas zu bestätigen. Bestätigen Sie das Abonnement.

3. Ersetzen Sie die dem Thema zugeordnete Zugriffsrichtlinie durch die folgende Richtlinie. Geben Sie darin Ihr SNS ThemaARN, Ihren Bucket-Namen und die Konto-ID des Bucket-Besitzers ein.

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "Example SNS topic policy",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "SNS-topic-ARN",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:bucket-name"
        },
        "StringEquals": {
          "aws:SourceAccount": "bucket-owner-account-id"
        }
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

4. Notieren Sie sich das ThemaARN.

Das von Ihnen erstellte SNS Thema ist eine weitere Ressource in Ihrem AWS-Konto, und es hat eine einzigartigeARN. Sie werden dies ARN im nächsten Schritt benötigen. Das ARN wird das folgende Format haben:

```
arn:aws:sns:aws-region:account-id:topic-name
```

Schritt 3: Hinzufügen einer Benachrichtigungskonfiguration zu Ihrem Bucket

Sie können Bucket-Benachrichtigungen entweder mithilfe der Amazon S3 S3-Konsole oder programmgesteuert aktivieren, indem Sie AWS SDKs. Wählen Sie eine der Optionen für die Konfiguration von Benachrichtigungen über Ihren Bucket. Dieser Abschnitt enthält Codebeispiele mit dem AWS SDKs für Java und .NET.

Option A: Aktivieren von Benachrichtigungen über einen Bucket unter Verwendung der Konsole

Fügen Sie mithilfe der Amazon-S3-Konsole eine Benachrichtigungskonfiguration hinzu, die Amazon S3 zu Folgendem auffordert:

- Veröffentlichen Sie Ereignisse des Typs Alle Objekte erstellen in Ihrer SQS Amazon-Warteschlange.
- Veröffentlichen Sie Ereignisse des Typs Object in RRS Lost in Ihrem SNS Amazon-Thema.

Nachdem Sie die Benachrichtigungskonfiguration gespeichert haben, veröffentlicht Amazon S3 eine Testmeldung, die Sie per E-Mail erhalten.

Detaillierte Anweisungen finden Sie unter [Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole](#).

Option B: Aktivieren Sie Benachrichtigungen für einen Bucket mithilfe der AWS SDKs

.NET

Das folgende C#-Codebeispiel zeigt ein vollständiges Listing, das einem Bucket eine Benachrichtigungskonfiguration hinzufügt. Sie müssen den Code aktualisieren und Ihren Bucket-

Namen und Ihr SNS Thema angebenARN. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class EnableNotificationsTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string snsTopic = "**** SNS topic ARN ****";
        private const string sqsQueue = "**** SQS topic ARN ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            EnableNotificationAsync().Wait();
        }

        static async Task EnableNotificationAsync()
        {
            try
            {
                PutBucketNotificationRequest request = new
PutBucketNotificationRequest
                {
                    BucketName = bucketName
                };

                TopicConfiguration c = new TopicConfiguration
                {
                    Events = new List<EventType> { EventType.ObjectCreatedCopy },
                    Topic = snsTopic
                };
            }
        }
    }
}
```



```
        });
        request.TopicConfigurations = new List<TopicConfiguration>();
        request.TopicConfigurations.Add(c);
        request.QueueConfigurations = new List<QueueConfiguration>();
        request.QueueConfigurations.Add(new QueueConfiguration()
        {
            Events = new List<EventType> { EventType.ObjectCreatedPut },
            Queue = sqsQueue
        });

        PutBucketNotificationResponse response = await
client.PutBucketNotificationAsync(request);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' ",
e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown error encountered on server.
Message:'{0}' ", e.Message);
    }
}
}
}
```

Java

Die folgende Beispiel veranschaulicht, wie Sie einem Bucket eine Benachrichtigungskonfiguration hinzufügen. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
```

```
import java.util.EnumSet;

public class EnableNotificationOnABucket {

    public static void main(String[] args) throws IOException {
        String bucketName = "**** Bucket name ****";
        Regions clientRegion = Regions.DEFAULT_REGION;
        String snsTopicARN = "**** SNS Topic ARN ****";
        String sqsQueueARN = "**** SQS Queue ARN ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .build();

            BucketNotificationConfiguration notificationConfiguration = new
BucketNotificationConfiguration();

            // Add an SNS topic notification.
            notificationConfiguration.addConfiguration("snsTopicConfig",
                new TopicConfiguration(snsTopicARN,
EnumSet.of(S3Event.ObjectCreated)));

            // Add an SQS queue notification.
            notificationConfiguration.addConfiguration("sqsQueueConfig",
                new QueueConfiguration(sqsQueueARN,
EnumSet.of(S3Event.ObjectCreated)));

            // Create the notification configuration request and set the bucket
notification
            // configuration.
            SetBucketNotificationConfigurationRequest request = new
SetBucketNotificationConfigurationRequest(
                bucketName, notificationConfiguration);
            s3Client.setBucketNotificationConfiguration(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

Schritt 4: Testen der Einrichtung

Jetzt können Sie das Setup testen, indem Sie ein Objekt in Ihren Bucket hochladen und die Ereignisbenachrichtigung in der SQS Amazon-Konsole überprüfen. Anweisungen finden Sie unter [Empfangen einer Nachricht](#) im Abschnitt "Erste Schritte" im Amazon Simple Queue Service-Entwicklerhandbuch.

Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung

Bei der Konfiguration einer Amazon-S3-Ereignis-Benachrichtigung müssen Sie angeben, welche unterstützten Amazon-S3-Ereignistypen das Senden der Benachrichtigung durch Amazon S3 veranlassen. Wenn ein Ereignistyp, den Sie nicht angegeben haben, in Ihrem S3-Bucket auftritt, sendet Amazon S3 die Benachrichtigung nicht.

Sie können Benachrichtigungen konfigurieren, die nach dem Präfix und dem Suffix des Objektschlüsselnamens gefiltert werden. Sie können beispielsweise eine Konfiguration so einrichten, dass Sie nur eine Benachrichtigung erhalten, wenn einem Bucket Bilddateien mit der Erweiterung „.jpg“ hinzugefügt werden. Oder Sie können eine Konfiguration einrichten, die eine Benachrichtigung an ein SNS Amazon-Thema sendet, wenn ein Objekt mit dem Präfix `images/` zum Bucket hinzugefügt wird, während Benachrichtigungen für Objekte mit dem Präfix `logs/` im selben Bucket an ein AWS Lambda Funktion.

Note

Ein Platzhalterzeichen (*) kann in Filtern nicht als Präfix oder Suffix verwendet werden. Wenn Ihr Präfix oder Suffix ein Leerzeichen enthält, müssen Sie es durch das Zeichen „+“ ersetzen. Wenn Sie weitere Sonderzeichen im Wert des Präfix oder Suffixes verwenden, müssen Sie diese im Format [URL-kodiert \(prozentkodiert\)](#) eingeben. Eine vollständige Liste der Sonderzeichen, die in ein URL -kodiertes Format konvertiert werden müssen, wenn sie in einem Präfix oder Suffix für Ereignisbenachrichtigungen verwendet werden, finden Sie unter [Sichere Zeichen](#)

Sie können Benachrichtigungskonfigurationen einrichten, die die Filterung von Objektschlüsselnamen in der Amazon-S3-Konsole verwenden. Sie können dies tun, indem Sie Amazon S3 APIs über den AWS SDKs oder das REST APIs direkt. Informationen zur Verwendung der Konsolen-Benutzeroberfläche zum Festlegen einer Benachrichtigungskonfiguration für einen Bucket finden Sie unter [Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole](#).

Amazon S3 speichert die Benachrichtigungskonfiguration wie XML in der Unterressource für Benachrichtigungen, die einem Bucket zugeordnet ist, wie unter beschrieben. [Verwendung von Amazon SQS SNS, Amazon und Lambda](#) Sie verwenden die Filter XML Struktur, um die Regeln für Benachrichtigungen zu definieren, die nach dem Präfix oder Suffix eines Objektschlüsselnamens gefiltert werden sollen. Informationen zur Filter XML Struktur finden Sie unter [PUT Bucket-Benachrichtigung](#) in der Amazon Simple Storage Service API Reference.

Benachrichtigungskonfigurationen, die Filter verwenden, können keine Filterregeln mit überlappenden Präfixen, überlappenden Suffixen oder Präfix- und Suffix-Überlappung definieren. Die folgenden Abschnitte enthalten Beispiele für gültige Benachrichtigungskonfigurationen mit Objektschlüssel-Namensfilterung. Sie enthalten auch Beispiele für Benachrichtigungskonfigurationen, die wegen der Überlappung von Präfix und Suffix ungültig sind.

Themen

- [Beispiele für gültige Benachrichtigungskonfigurationen mit Filterung nach dem Objektschlüsselnamen](#)
- [Beispiele für Benachrichtigungskonfigurationen mit ungültiger Präfix- und Suffix-Überlappung](#)

Beispiele für gültige Benachrichtigungskonfigurationen mit Filterung nach dem Objektschlüsselnamen

Die folgende Benachrichtigungskonfiguration enthält eine Warteschlangenkonfiguration, die eine SQS Amazon-Warteschlange identifiziert, in der Amazon S3 Ereignisse dieses `s3:ObjectCreated:Put` Typs veröffentlichen kann. Die Ereignisse werden immer dann veröffentlicht, wenn sich ein Objekt mit dem Präfix `images/` und einem `jpg` Suffix in einem Bucket befindet `PUT`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
```

```

        <Value>images/</Value>
    </FilterRule>
    <FilterRule>
        <Name>suffix</Name>
        <Value>jpg</Value>
    </FilterRule>
</S3Key>
</Filter>
<Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
<Event>s3:ObjectCreated:Put</Event>
</QueueConfiguration>
</NotificationConfiguration>

```

Die folgende Benachrichtigungskonfigurationen hat mehrere nicht überlappende Präfixe. Die Konfiguration definiert, dass Benachrichtigungen für PUT Anfragen im `images/` Ordner in Warteschlange A und Benachrichtigungen für PUT Anfragen im `logs/` Ordner in Warteschlange B verschoben werden.

```

<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images/</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-A</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
  <QueueConfiguration>
    <Id>2</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>logs/</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-B</Queue>
  </QueueConfiguration>
</NotificationConfiguration>

```

```

    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>

```

Die folgende Benachrichtigungskonfigurationen hat mehrere nicht überlappende Suffixe. Die Konfiguration definiert, dass alle dem Bucket neu hinzugefügten .jpg-Bilder über Lambda cloud-function-A und alle neu hinzugefügten .png-Bilder über cloud-function-B verarbeitet werden. Die .png und .jpg Suffixe überschneiden sich nicht, obwohl sie den gleichen Endbuchstaben haben. Wenn eine bestimmte Zeichenfolge mit beiden Suffixen enden kann, werden die beiden Suffixe als überlappend betrachtet. Da eine Zeichenfolge nicht mit .png und .jpg gleichzeitig enden kann, sind die Suffixe in der Beispielkonfiguration keine überlappenden Suffixe.

```

<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
    <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
    <Id>2</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.png</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
    <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>

```

Ihre Benachrichtigungskonfigurationen, die Filter verwenden, können keine Filterregeln mit überlappenden Präfixen für dieselben Ereignistypen definieren. Sie können dies nur tun, wenn die sich überlappenden Präfixe, die mit Suffixen verwendet werden, die sich nicht überlappen. Die folgende Beispielkonfiguration zeigt, wie Objekte, die mit einem gemeinsamen Präfix, aber nicht überlappenden Suffixen erstellt werden, an unterschiedliche Ziele geliefert werden können.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
    <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
    <Id>2</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.png</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
    <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
```

```
</NotificationConfiguration>
```

Beispiele für Benachrichtigungskonfigurationen mit ungültiger Präfix- und Suffix-Überlappung

Ihre Benachrichtigungskonfigurationen, die `Filter` verwenden, können größtenteils keine Filterregeln mit überlappenden Präfixen, überlappenden Suffixen oder überlappenden Präfix- und Suffix-Kombinationen für dieselben Ereignistypen definieren. Überlappende Präfixe sind möglich, solange sich die Suffixe nicht überlappen. Ein Beispiel finden Sie unter [Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung](#).

Sie können überlappende Objektschlüsselnamensfilter mit unterschiedlichen Ereignistypen verwenden. Beispielsweise könnten sie eine Benachrichtigungskonfiguration erstellen, die das Präfix `image/` für den Ereignistyp `ObjectCreated:Put` und das Präfix `image/` für den Ereignistyp `ObjectRemoved:*` verwendet.

Sie erhalten eine Fehlermeldung, wenn Sie versuchen, eine Benachrichtigungskonfiguration zu speichern, die ungültige überlappende Namensfilter für dieselben Ereignistypen enthält, wenn Sie die Amazon S3 S3-Konsole verwenden oder API. Dieser Abschnitt zeigt Beispiele für Benachrichtigungskonfigurationen, die aufgrund überlappender Namensfilter ungültig sind.

Es wird angenommen, dass eine vorhandene Benachrichtigungskonfigurationsregel ein Standardpräfix und -suffix hat, die mit allen anderen Präfixen bzw. Suffixen übereinstimmen. Die folgende Benachrichtigungskonfiguration ist ungültig, weil sie überlappende Präfixe hat. Insbesondere überlappt sich das Root-Präfix mit allen anderen Präfixen. Das gleiche gilt, wenn Sie in diesem Beispiel ein Suffix anstelle eines Präfixes verwenden. Das Root-Suffix überlappt mit allen anderen Suffixen.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-one</Topic>
    <Event>s3:ObjectCreated:*</Event>
  </TopicConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-two</Topic>
    <Event>s3:ObjectCreated:*</Event>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images</Value>
        </FilterRule>
      </S3Key>
    </Filter>
  </TopicConfiguration>
</NotificationConfiguration>
```



```

    </S3Key>
  </Filter>
</TopicConfiguration>
</NotificationConfiguration>

```

Die folgende Benachrichtigungskonfiguration ist ungültig, weil sie überlappende Suffixe hat. Wenn eine bestimmte Zeichenfolge mit beiden Suffixen enden kann, werden die beiden Suffixe als überlappend betrachtet. Eine Zeichenfolge kann mit jpg und pg enden. Die Suffixe überschneiden sich also. Gleiches gilt für Präfixe. Wenn eine bestimmte Zeichenfolge mit beiden Präfixen beginnen kann, werden die beiden Präfixe als überlappend betrachtet.

```

<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
    <Event>s3:ObjectCreated:*</Event>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>suffix</Name>
          <Value>jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
  </TopicConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
    <Event>s3:ObjectCreated:Put</Event>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>suffix</Name>
          <Value>pg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
  </TopicConfiguration>
</NotificationConfiguration>

```

Die folgende Benachrichtigungskonfiguration ist ungültig, weil sie überlappende Präfixe und Suffixe hat.

```

<NotificationConfiguration>

```

```
<TopicConfiguration>
  <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
  <Event>s3:ObjectCreated:*</Event>
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>prefix</Name>
        <Value>images</Value>
      </FilterRule>
      <FilterRule>
        <Name>suffix</Name>
        <Value>jpg</Value>
      </FilterRule>
    </S3Key>
  </Filter>
</TopicConfiguration>
<TopicConfiguration>
  <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
  <Event>s3:ObjectCreated:Put</Event>
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>suffix</Name>
        <Value>jpg</Value>
      </FilterRule>
    </S3Key>
  </Filter>
</TopicConfiguration>
</NotificationConfiguration>
```

Struktur von Ereignismeldungen

Die Benachrichtigungsnachricht, die Amazon S3 zur Veröffentlichung eines Ereignisses sendet, hat das folgende JSON Format.

Eine allgemeine Übersicht und Anweisungen zum Konfigurieren von Ereignisbenachrichtigungen finden Sie unter [Amazon-S3-Ereignis-Benachrichtigungen](#).

Dieses Beispiel zeigt Version 2.2 der JSON Struktur der Ereignisbenachrichtigung. Amazon S3 verwendet Versionen 2.1 und 2.2 und 2.3 dieser Ereignisstruktur. Amazon S3 verwendet Version 2.2 für regionsübergreifende Replikationsereignisbenachrichtigungen. Es verwendet Version 2.3 für Ereignisse wie S3 Lifecycle, S3 Intelligent-TieringACL, Objekt- und Objekt-Tagging und

Objektwiederherstellung zum Löschen von Objekten. Diese Versionen enthalten zusätzliche Informationen, die für diese Vorgänge spezifisch sind. Die Versionen 2.2 und 2.3 sind ansonsten mit Version 2.1 kompatibel, die Amazon S3 derzeit für alle anderen Ereignisbenachrichtigungstypen verwendet.

```
{
  "Records": [
    {
      "eventVersion": "2.2",
      "eventSource": "aws:s3",
      "awsRegion": "us-west-2",
      "eventTime": "The time, in ISO-8601 format, for example,
1970-01-01T00:00:00.000Z, when Amazon S3 finished processing the request",
      "eventName": "event-type",
      "userIdentity": {
        "principalId": "Amazon-customer-ID-of-the-user-who-caused-the-event"
      },
      "requestParameters": {
        "sourceIPAddress": "ip-address-where-request-came-from"
      },
      "responseElements": {
        "x-amz-request-id": "Amazon S3 generated request ID",
        "x-amz-id-2": "Amazon S3 host that processed the request"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "ID found in the bucket notification configuration",
        "bucket": {
          "name": "bucket-name",
          "ownerIdentity": {
            "principalId": "Amazon-customer-ID-of-the-bucket-owner"
          },
          "arn": "bucket-ARN"
        },
        "object": {
          "key": "object-key",
          "size": "object-size in bytes",
          "eTag": "object eTag",
          "versionId": "object version if bucket is versioning-enabled, otherwise
null",
          "sequencer": "a string representation of a hexadecimal value used to
determine event sequence, only used with PUTs and DELETes"
        }
      }
    }
  ]
}
```

```
    },
    "glacierEventData": {
      "restoreEventData": {
        "lifecycleRestorationExpiryTime": "The time, in ISO-8601 format, for
example, 1970-01-01T00:00:00.000Z, of Restore Expiry",
        "lifecycleRestoreStorageClass": "Source storage class for restore"
      }
    }
  }
]
```

Beachten Sie im Zusammenhang mit der Ereignistachrichtenstruktur Folgendes:

- Der `eventVersion`-Schlüsselwert enthält eine Haupt- und eine Nebenversion im Format `<major>.<minor>`.

Die Hauptversion wird erhöht, wenn Amazon S3 eine Änderung an der Ereignisstruktur vornimmt, die nicht abwärtskompatibel ist. Dazu gehört das Entfernen eines bereits vorhandenen JSON Felds oder das Ändern der Darstellung des Feldinhalts (z. B. ein Datumsformat).

Die Nebenversion wird erhöht, wenn Amazon S3 der Ereignisstruktur neue Felder hinzufügt. Dies kann auftreten, wenn neue Informationen für einige oder alle vorhandenen Ereignisse bereitgestellt werden. Dies kann auch geschehen, wenn neue Informationen nur zu neu eingeführten Ereignistypen bereitgestellt werden. Anwendungen sollten neue Felder ignorieren, um weiter aufwärtskompatibel mit neuen Nebenversionen der Ereignisstruktur zu sein.

Falls neue Ereignistypen eingeführt werden, aber die Struktur des Ereignisses anderweitig unmodifiziert ist, ändert sich die Ereignisversion nicht.

Um sicherzustellen, dass Ihre Anwendungen die Ereignisstruktur ordnungsgemäß analysieren können, empfehlen wir, dass Sie einen Vergleich mit der Hauptversionsnummer durchführen. Um sicherzustellen, dass die Felder, die von Ihrer Anwendung erwartet werden, vorhanden sind, empfehlen wir außerdem, mit der Nebenversion einen `greater-than-or-equal` -to-Vergleich durchzuführen.

- Die `eventName` verweist auf die Liste der [Typen der Ereignisbenachrichtigung](#), enthält aber nicht die `s3:-`Präfix.
- Der `responseElements` Schlüsselwert ist nützlich, wenn Sie eine Anfrage nachverfolgen möchten, indem Sie Folgendes eingeben AWS Support. Beides `x-amz-request-id` und `x-amz-id-2` hilft Amazon S3 dabei, eine einzelne Anfrage nachzuverfolgen. Diese Werte sind dieselben,

die Amazon S3 in der Antwort auf die Anforderung zurückgibt, die die Ereignisse initiiert. Auf diese Weise können sie verwendet werden, um das Ereignis der Anfrage zuzuordnen.

- Der `s3`-Schlüssel bietet Informationen über den Bucket und das Objekt, die an dem Ereignis beteiligt sind. Der Wert des Objektschlüsselnamens ist URL codiert. Beispielsweise wird "red flower.jpg" zu "red+flower.jpg" (Amazon S3 gibt als Inhaltstyp in der Antwort "application/x-www-form-urlencoded" zurück).
- Der `sequenceNumber`-Schlüssel bietet eine Möglichkeit, die Reihenfolge von Ereignissen zu bestimmen. Ereignis-Benachrichtigungen kommen nicht garantiert in der Reihenfolge an, in der die Ereignisse aufgetreten sind. Benachrichtigungen von Ereignissen, die Objekte erstellen (PUT) und Objekte löschen, enthalten jedoch ein `sequenceNumber`. Es kann verwendet werden, um die Reihenfolge der Ereignisse für einen bestimmten Objektschlüssel zu bestimmen.

Wenn Sie die `sequenceNumber`-Zeichenfolgen von zwei Ereignisbenachrichtigungen für denselben Objektschlüssel vergleichen, ist die Ereignisbenachrichtigung mit dem größeren hexadezimalen Wert von `sequenceNumber` das später aufgetretene Ereignis. Wenn Sie Ereignisbenachrichtigungen verwenden, um eine separate Datenbank oder einen separaten Index Ihrer Amazon-S3-Objekte zu verwalten, empfehlen wir Ihnen, die `sequenceNumber`-Werte während der Verarbeitung jeder Ereignisbenachrichtigung zu vergleichen und zu speichern.

Beachten Sie Folgendes:

- Sie können `sequenceNumber` nicht verwenden, um die Reihenfolge von Ereignissen für unterschiedliche Objektschlüssel zu bestimmen.
- Die Sequenzer können unterschiedliche Längen haben. Um diese Werte zu vergleichen, füllen Sie zuerst den kürzeren Wert links mit Nullen auf und führen dann einen alphabetischen Vergleich durch.
- Der `glacierEventData`-Schlüssel ist nur für `s3:ObjectRestore:Completed`-Ereignisse sichtbar.
- Der `restoreEventData`-Schlüssel enthält Attribute, die sich auf Ihre Wiederherstellungsanfrage beziehen.
- Der Schlüssel `replicationEventData` ist nur für Replikationsereignisse sichtbar.
- Der `intelligentTieringEventData`-Schlüssel ist nur für Ereignisse von S3 Intelligent Tiering sichtbar.
- Der `lifecycleEventData`-Schlüssel ist nur für S3-Lebenszyklus-Übergangereignisse sichtbar.

Beispielnachrichten

Im Folgenden finden Sie Beispiele für Amazon-S3-Ereignis-Benachrichtigungen.

Amazon-S3-Testnachricht

Nachdem Sie eine Ereignis-Benachrichtigung für einen Bucket konfigurieren, sendet Amazon S3 die folgende Testnachricht.

```
{
  "Service": "Amazon S3",
  "Event": "s3:TestEvent",
  "Time": "2014-10-13T15:57:02.089Z",
  "Bucket": "bucketname",
  "RequestId": "5582815E1AEA5ADF",
  "HostId": "8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wrR+YstdA6Knx4Ip8EXAMPLE"
}
```

Beispielnachricht, wenn ein Objekt mithilfe einer PUT Anfrage erstellt wird

Die folgende Nachricht ist ein Beispiel für eine Nachricht, die Amazon S3 zum Veröffentlichen eines `s3:ObjectCreated:Put`-Ereignisses sendet.

```
{
  "Records": [
    {
      "eventVersion": "2.1",
      "eventSource": "aws:s3",
      "awsRegion": "us-west-2",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "AIDAJDPLRKL7UEXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "C3D13FE58DE4C810",
        "x-amz-id-2": "FMYUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvAN0jpD"
      },
      "s3": {
```

```

    "s3SchemaVersion": "1.0",
    "configurationId": "testConfigRule",
    "bucket": {
      "name": "mybucket",
      "ownerIdentity": {
        "principalId": "A3NL1K0ZZKExample"
      },
      "arn": "arn:aws:s3:::mybucket"
    },
    "object": {
      "key": "HappyFace.jpg",
      "size": 1024,
      "eTag": "d41d8cd98f00b204e9800998ecf8427e",
      "versionId": "096fKKXTRTt13on89fV0.nfljtsv6qko",
      "sequencer": "0055AED6DCD90281E5"
    }
  }
}
]
}

```

Eine Definition der einzelnen IAM Identifikationspräfixe (z. B. AIDA, AROA, AGPA) finden Sie im IAM Benutzerhandbuch unter [IAM Identifikatoren](#).

Verwenden EventBridge

Amazon S3 kann Ereignisse an Amazon EventBridge senden, wenn bestimmte Ereignisse in Ihrem Bucket auftreten. Im Gegensatz zu anderen Zielen müssen Sie nicht auswählen, welche Ereignistypen Sie liefern möchten. Nach der Aktivierung von EventBridge werden alle unten aufgeführten Ereignisse an EventBridge gesendet. Sie können EventBridge Regeln verwenden, um Ereignisse an weitere Ziele weiterzuleiten. Im Folgenden sind die Ereignisse aufgeführt, an die Amazon S3 EventBridge sendet.

Ereignistyp	Beschreibung
Objekt erstellt	<p>Ein Objekt wurde erstellt.</p> <p>Das Feld „Grund“ in der Struktur der Ereignisnachricht gibt an, welches S3 zur Erstellung des Objekts verwendet wurde: PutObject, POSTObject, CopyObject, oder CompleteMultipartUpload.</p>

Ereignistyp	Beschreibung
Objekt gelöscht (DeleteObject)	Ein Objekt wurde gelöscht.
Objekt gelöscht (Lebenszyklusablauf)	<p>Wenn ein Objekt mithilfe eines API S3-Aufrufs gelöscht wird, ist das Feld für den Grund auf gesetzt DeleteObject. Wenn ein Objekt durch eine S3-Lebenszyklus-Ablaufregel gelöscht wird, wird das Grundfeld auf Lebenszyklus-Ablauf gesetzt. Weitere Informationen finden Sie unter Auslaufen de Objekte.</p> <p>Wenn ein nicht versioniertes Objekt gelöscht wird oder ein versioniertes Objekt dauerhaft gelöscht wird, wird das Feld für den Löschtyp auf Dauerhaft gelöscht gesetzt. Wenn eine Löschemarkierung für ein versioniertes Objekt erstellt wird, wird das Feld für den Löschtyp auf Löschemarkierung erstellt gesetzt. Weitere Informationen finden Sie unter Löschen von Objekten aus einem versioning-fähigen Bucket.</p>
Objektwiederherstellung eingeleitet	Eine Objektwiederherstellung wurde von der Speicherklasse S3 Glacier oder S3 Glacier Deep Archive oder von der Ebene S3 Intelligent-Tiering Archive Access oder Deep Archive Access initiiert. Weitere Informationen finden Sie unter Arbeiten mit archivierten Objekten .
Objektwiederherstellung abgeschlossen	Eine Objektwiederherstellung wurde abgeschlossen.
Wiederherstellen von Objekten ist abgelaufen	Die temporäre Kopie eines Objekts, das aus S3 Glacier oder S3 Glacier Deep Archive wiederhergestellt wurde, abgelaufen und wurde gelöscht.
Objektspeicherklasse geändert	Ein Objekt wurde auf eine andere Speicherklasse umgestellt. Weitere Informationen finden Sie unter Übergang von Objekten mit Amazon-S3-Lebenszyklus .

Ereignistyp	Beschreibung
Objektzugriffsebene wurde geändert	Ein Objekt wurde auf die Stufen S3 Intelligent-Tiering Archive Access oder Deep Archive Access umgestellt. Weitere Informationen finden Sie unter Amazon S3 Intelligent Tiering .
Objekt ACL wurde aktualisiert	Die Zugriffskontrollliste (ACL) eines Objekts wurde mit gesetzt PutObjectACL. Ein Ereignis wird nicht generiert, wenn eine Anforderung zu keiner Änderung an der eines Objekts führtACL. Weitere Informationen finden Sie unter Übersicht über die Zugriffskontrollliste (ACL) .
Objekt-Tags hinzugefügt	Ein Satz von Tags wurde einem Objekt hinzugefügt mit PutObjectTagging. Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen .
Gelöschte Objekt-Tags	Alle Tags wurden mithilfe von aus einem Objekt entfernt DeleteObjectTagging. Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen .

Note

Weitere Informationen darüber, wie Amazon S3 S3-Ereignistypen EventBridge Ereignistypen zugeordnet werden, finden Sie unter [EventBridge Amazon-Kartierung und Problembehebung](#).

Sie können Amazon S3 Event Notifications mit verwenden EventBridge , um Regeln zu schreiben, die Maßnahmen ergreifen, wenn ein Ereignis in Ihrem Bucket eintritt. Sie können sich beispielsweise eine Benachrichtigung senden lassen. Weitere Informationen finden Sie unter [Was steht EventBridge](#) im EventBridge Amazon-Benutzerhandbuch.

Weitere Informationen zu den Aktionen und Datentypen, mit denen Sie interagieren können EventBridge API, finden Sie in der [EventBridge APIAmazon-Referenz](#) in der EventBridge APIAmazon-Referenz.

Informationen zu den Preisen finden Sie unter [EventBridge Amazon-Preise](#).

Themen

- [EventBridge Amazon-Berechtigungen](#)
- [Amazon aktivieren EventBridge](#)
- [EventBridge Struktur der Ereignisnachricht](#)
- [EventBridge Amazon-Kartierung und Problembehebung](#)

EventBridge Amazon-Berechtigungen

Amazon S3 benötigt keine zusätzlichen Berechtigungen, um Ereignisse an Amazon zu übermitteln EventBridge.

Amazon aktivieren EventBridge

Sie können Amazon EventBridge über die S3-Konsole aktivieren, AWS Command Line Interface (AWS CLI) oder Amazon S3 RESTAPI.

Verwenden der S3-Konsole

Um die Übertragung von EventBridge Ereignissen in der S3-Konsole zu aktivieren.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Ereignisse aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Gehen Sie zum Abschnitt Event-Benachrichtigungen und suchen Sie den EventBridgeAmazon-Unterabschnitt. Wählen Sie Edit (Bearbeiten) aus.
5. Wählen Sie unter Benachrichtigungen EventBridge für alle Ereignisse in diesem Bucket an Amazon senden die Option An.

Note

Nach der Aktivierung dauert es etwa fünf Minuten EventBridge, bis die Änderungen wirksam werden.

Verwendung der AWS CLI

Im folgenden Beispiel wird eine Bucket-Benachrichtigungskonfiguration für einen Bucket `amzn-s3-demo-bucket1` mit EventBridge aktiviertem Amazon erstellt.

```
aws s3api put-bucket-notification-configuration --bucket amzn-s3-demo-bucket1 --notification-configuration='{ "EventBridgeConfiguration": {} }'
```

Mit dem REST API

Sie können Amazon EventBridge programmgesteuert in einem Bucket aktivieren, indem Sie Amazon S3 aufrufen. REST API Weitere Informationen finden Sie unter [PutBucketNotificationConfiguration](#) Amazon Simple Storage Service API Reference.

Das folgende Beispiel zeigt die Konfiguration, die XML verwendet wird, um eine Bucket-Benachrichtigung zu erstellen, wenn Amazon EventBridge aktiviert ist.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <EventBridgeConfiguration>
  </EventBridgeConfiguration>
</NotificationConfiguration>
```

EventBridge Regeln erstellen

Nach der Aktivierung können Sie EventBridge Amazon-Regeln für bestimmte Aufgaben erstellen. Beispielsweise können Sie E-Mail-Benachrichtigungen senden, wenn ein Objekt erstellt wird. Ein vollständiges Tutorial finden Sie unter [Tutorial: Eine Benachrichtigung senden, wenn ein Amazon S3 S3-Objekt erstellt wird](#) im EventBridge Amazon-Benutzerhandbuch.

EventBridge Struktur der Ereignisnachricht

Die Benachrichtigungsnachricht, die Amazon S3 zur Veröffentlichung eines Ereignisses sendet, hat das folgende JSON Format. Wenn Amazon S3 ein Ereignis an Amazon sendet EventBridge, sind die folgenden Felder vorhanden.

- Ausführung – Derzeit 0 (Null) für alle Ereignisse.
- id — Eine Version 4, die für jedes Ereignis UUID generiert wird.
- detail-type – Die Art des Ereignisses, das gesendet wird. Eine Liste der Ereignistypen finden Sie unter [Verwenden EventBridge](#).

- Quelle – Gibt den Service an, aus dem das Ereignis stammt.
- Konto — Das 12-stellige AWS-Konto ID des Bucket-Besitzers.
- Zeit – Die Zeit, zu der das Ereignis aufgetreten ist.
- Region — Identifiziert die AWS-Region des Buckets.
- resources — Ein JSON Array, das den Amazon-Ressourcennamen (ARN) des Buckets enthält.
- detail — Ein JSON Objekt, das Informationen über das Ereignis enthält. Weitere Informationen dazu, was in diesem Feld enthalten sein kann, finden Sie unter [Detailfeld für Ereignismeldung](#).

Beispiele für Ereignismeldungen

Im Folgenden finden Sie Beispiele für einige der Amazon S3 S3-Ereignisbenachrichtigungen, die an Amazon gesendet werden können EventBridge.

Objekt erstellt

```
{
  "version": "0",
  "id": "17793124-05d4-b198-2fde-7ededc63b103",
  "detail-type": "Object Created",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "IYV3p45BT0ac8hjHg1houSdS1a.Mro8e",
      "sequencer": "617f08299329d189"
    }
  },
  "request-id": "N4N7GDK58NMKJ12R",
  "requester": "123456789012",
```

```
"source-ip-address": "1.2.3.4",
"reason": "PutObject"
}
}
```

Objekt wurde gelöscht (mithilfe von DeleteObject)

```
{
  "version": "0",
  "id": "2ee9cc15-d022-99ea-1fb8-1b1bac4850f9",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "1QW9g1Z99LUNbvaaYVpW9xD10LU.qxgF",
      "sequencer": "617f0837b476e463"
    },
    "request-id": "0BH729840619AG5K",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "DeleteObject",
    "deletion-type": "Delete Marker Created"
  }
}
```

Objekt wurde gelöscht (unter Verwendung des Lebenszyklusablaufs)

```
{
  "version": "0",
```

```
"id": "ad1de317-e409-eba2-9552-30113f8d88e3",
"detail-type": "Object Deleted",
"source": "aws.s3",
"account": "111122223333",
"time": "2021-11-12T00:00:00Z",
"region": "ca-central-1",
"resources": [
  "arn:aws:s3:::amzn-s3-demo-bucket1"
],
"detail": {
  "version": "0",
  "bucket": {
    "name": "amzn-s3-demo-bucket1"
  },
  "object": {
    "key": "example-key",
    "etag": "d41d8cd98f00b204e9800998ecf8427e",
    "version-id": "mtB0cV.jejK63XkRNceanNMC.qXPWLeK",
    "sequencer": "617b398000000000"
  },
  "request-id": "20EB74C14654DC47",
  "requester": "s3.amazonaws.com",
  "reason": "Lifecycle Expiration",
  "deletion-type": "Delete Marker Created"
}
}
```

Objektwiederherstellung abgeschlossen

```
{
  "version": "0",
  "id": "6924de0d-13e2-6bbf-c0c1-b903b753565e",
  "detail-type": "Object Restore Completed",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
```

```
"bucket": {
  "name": "amzn-s3-demo-bucket1"
},
"object": {
  "key": "example-key",
  "size": 5,
  "etag": "b1946ac92492d2347c6235b4d2611184",
  "version-id": "KKsjUC1.6gIjqtvhfg5AdMI0eCePIiT3"
},
"request-id": "189F19CB7FB1B6A4",
"requester": "s3.amazonaws.com",
"restore-expiry-time": "2021-11-13T00:00:00Z",
"source-storage-class": "GLACIER"
}
}
```

Detailfeld für Ereignismeldung

Das Detailfeld enthält ein JSON Objekt mit Informationen über das Ereignis. Die folgenden Felder können im Detailfeld vorhanden sein.

- Ausführung – Derzeit 0 (Null) für alle Ereignisse.
- Bucket – Informationen über den Amazon-S3-Bucket, der an der Veranstaltung beteiligt ist.
- Objekt – Informationen über das an dem Ereignis beteiligte Amazon-S3-Objekt.
- request-id – Anforderungs-ID in S3 Antwort.
- Anforderer — AWS-Konto ID oder AWS Servicechef des Anfragenden.
- source-ip-address— Quell-IP-Adresse der S3-Anfrage. Nur vorhanden für Ereignisse, die durch eine S3-Anfrage ausgelöst werden.
- Grund — Für Object Created-Ereignisse wurde das S3 API verwendet, um das Objekt zu erstellen: [PutObject](#), [POSTObject CopyObject](#), oder [CompleteMultipartUpload](#). DeleteObjectBei Ereignissen mit gelöschtem Objekt ist dies so eingestellt, dass ein Objekt durch einen API S3-Aufruf gelöscht wird, oder Lifecycle Expiration, wenn ein Objekt durch eine S3-Lebenszyklus-Ablaufregel gelöscht wird. Weitere Informationen finden Sie unter [Auslaufende Objekte](#).
- deletion-type – Wenn ein nicht versioniertes Objekt gelöscht wird oder ein versioniertes Objekt dauerhaft gelöscht wird, wird bei Object Deleted-Ereignissen dies auf Permanently Deleted festgelegt. Wenn ein Löschmarker für ein versioniertes Objekt erstellt wird, wird dieser auf Löschmarkierung erstellt gesetzt. Weitere Informationen finden Sie unter [Löschen von Objekten aus einem versioning-fähigen Bucket](#).

Note

Einige Objektattribute (wie `etag` und `size`) sind nur vorhanden, wenn eine Löschmarke erstellt wird.

- `restore-expiry-time`— Bei Ereignissen mit Abschluss der Objektwiederherstellung ist dies der Zeitpunkt, zu dem die temporäre Kopie des Objekts aus S3 gelöscht wird. Weitere Informationen finden Sie unter [Arbeiten mit archivierten Objekten](#).
- `source-storage-class`— Bei Ereignissen wie „Object Restore Initiated“ und „Object Restore Completed“ die Speicherklasse des wiederherzustellenden Objekts. Weitere Informationen finden Sie unter [Arbeiten mit archivierten Objekten](#).
- `destination-storage-class`— Bei Ereignissen vom Typ Object Storage Class Changed die neue Speicherklasse des Objekts. Weitere Informationen finden Sie unter [Übergang von Objekten mit Amazon-S3-Lebenszyklus](#).
- `destination-access-tier`— Bei Ereignissen mit geänderter Objektzugriffsebene die neue Zugriffsebene des Objekts. Weitere Informationen finden Sie unter [Amazon S3 Intelligent Tiering](#).

EventBridge Amazon-Kartierung und Problembhebung

In der folgenden Tabelle wird beschrieben, wie Amazon S3 S3-Ereignistypen EventBridge Amazon-Ereignistypen zugeordnet werden.

S3-Ereignistyp	EventBridge Amazon-Detailtyp
ObjectCreatedAmazon-Detailtyp ----SEP----:PUT	Objekt erstellt
ObjectCreated:Put ----SEP----:Post	
ObjectCreated:Post ----SEP-- --:Kopieren	
ObjectCreated:CompleteMulti partUpload	
ObjectRemoved:Kopieren ----SEP-- --:Löschen	Objekt gelöscht

S3-Ereignistyp	EventBridge Amazon-Detailtyp
ObjectRemoved:DeleteMarkerCreated	
LifecycleExpiration:Löschen ---- SEP----:Löschen	
LifecycleExpiration:DeleteMarkerCreated	
ObjectRestore:Löschen ----SEP-- --:Beitrag	Objektwiederherstellung eingeleitet
ObjectRestore:Beitrag ----SEP-- --:Abgeschlossen	Objektwiederherstellung abgeschlossen
ObjectRestore:Abgeschlossen ---- SEP----:Löschen	Wiederherstellen von Objekten ist abgelaufen
LifecycleTransition	Objektspeicherklasse geändert
IntelligentTiering	Objektzugriffsebene wurde geändert
ObjectTagging:Löschen ----SEP-- --:PUT	Objekt-Tags hinzugefügt
ObjectTagging:Put ----SEP-- --:Löschen	Gelöschte Objekt-Tags
ObjectAcl:Löschen ----SEP----:PUT	Objekt wurde aktualisiert ACL

EventBridge Amazon-Problembhebung

Informationen zur Fehlerbehebung EventBridge finden Sie unter [Problembehandlung bei Amazon EventBridge](#) im EventBridge Amazon-Benutzerhandbuch.

Verwenden von Analysen und Einblicken

Sie können Analytik und Erkenntnisse in Amazon S3 verwenden, um Ihre Speichernutzung zu verstehen, zu analysieren und zu optimieren. Weitere Informationen finden Sie in den folgenden Themen.

Themen

- [Amazon S3 analytics – Speicherklassen-Analyse](#)
- [Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens](#)
- [Nachverfolgen von Amazon-S3-Anforderungen mit AWS X-Ray](#)

Amazon S3 analytics – Speicherklassen-Analyse

Mit der Speicherklassen-Analyse von Amazon S3 Analytics können Sie Speicherzugriffsmuster analysieren, anhand derer Sie entscheiden können, ob Sie die richtigen Daten in die richtige Speicherklasse einordnen. Diese neue analytische Funktion von Amazon S3 beobachtet Datenzugriffsmuster, anhand derer Sie entscheiden können, wann Sie STANDARD-Speicher mit weniger häufigem Zugriff in die Speicherklasse STANDARD_IA (IA für „infrequent access“ (seltener Zugriff)) überführen sollen. Weitere Informationen über Speicherklassen finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

Wenn die Speicherklassen-Analyse die seltenen Zugriffsmuster für eine gefilterte Datenmenge im Laufe der Zeit beobachtet, können Sie Ihre Lebenszykluskonfigurationen unter Verwendung der Analyseergebnisse verbessern. Sie können die Speicherklassen-Analyse so konfigurieren, dass alle Objekte in einem Bucket analysiert werden. Sie können aber auch Filter konfigurieren, um Objekte nach einem gemeinsamen Präfix zu gruppieren (d. h. für Objekte, deren Namen mit einer gemeinsamen Zeichenkette beginnen), nach Objekt-Markierungen oder sowohl nach Präfix und Markierungen. Wahrscheinlich werden Sie feststellen, dass die Filterung nach Objektgruppen die beste Methode darstellt, von der Speicherklassen-Analyse zu profitieren.

Important

Die Speicherklassenanalyse stellt nur Empfehlungen für die Klassen Standard bis Standard-IA bereit.

Sie können mehrere Speicherklassen-Analysefilter pro Bucket einrichten, bis zu 1000, und erhalten für jeden Filter eine separate Analyse. Durch Verwendung mehrerer Filterkonfigurationen können Sie spezifische Objektgruppen analysieren, um Ihre Lebenszykluskonfigurationen zu verbessern, die Objekte in STANDARD_IA überführen.

Die Speicherklassen-Analyse bietet Schaubilder zur Speichernutzung in der Amazon-S3-Konsole, die täglich aktualisiert werden. Sie können diese täglichen Nutzungsdaten auch in einen S3-Bucket exportieren und sie in einer Tabellenkalkulationsanwendung oder mit Business Intelligence-Tools wie Amazon QuickSight anzeigen.

Mit der Speicherklassen-Analyse sind Kosten verbunden. Preisinformationen finden Sie unter Verwaltung und Replikation [Amazon-S3-Preise](#).

Themen

- [Wie richte ich die Speicherklassen-Analyse ein?](#)
- [Wie verwende ich die Speicherklassen-Analyse?](#)
- [Wie kann ich die Daten der Speicherklassen-Analyse exportieren?](#)
- [Konfigurieren der Speicherklassen-Analyse](#)

Wie richte ich die Speicherklassen-Analyse ein?

Sie können die Speicherklassen-Analyse einrichten, indem Sie konfigurieren, welche Objektdaten analysiert werden sollen. Sie können die Speicherklassenanalyse für die folgenden Aufgaben konfigurieren:

- Analyse des gesamten Inhalts eines Buckets.

Sie erhalten eine Analyse für alle Objekte im Bucket.

- Analyse von nach Präfix und Tags gruppierten Objekten.

Sie können Filter konfigurieren, um Objekte nach einem gemeinsamen Präfix oder nach Objekt-Markierungen zu gruppieren, oder nach einer Kombination aus Präfix und Markierungen.

Sie erhalten für jeden konfigurierten Filter eine separate Analyse. Sie können maximal 1000 Filterkonfigurationen pro Bucket verwenden.

- Export von Analysedaten.

Wenn Sie die Speicherklassen-Analyse für einen Bucket oder Filter konfigurieren, können Sie die Analysedaten täglich in eine Datei exportieren lassen. Die Analyse für den Tag wird der Datei

hinzugefügt, um ein Verlaufsanalyseprotokoll für den konfigurierten Filter zu erstellen. Die Datei wird täglich an dem Ziel Ihrer Wahl aktualisiert. Wenn Sie Dateien zum Export auswählen, geben Sie einen Ziel-Bucket und ein optionales Ziel-Präfix an, wohin die Datei geschrieben werden soll.

Sie können die Amazon S3 S3-Konsole, die REST-API AWS CLI oder die AWS SDKs verwenden, um die Speicherklassenanalyse zu konfigurieren.

- Informationen zum Konfigurieren der Analyse von Speicherklassen in der Amazon-S3-Konsole finden Sie unter [Konfigurieren der Speicherklassen-Analyse](#).
- Um die Amazon S3 S3-API zu verwenden, verwenden Sie die [PutBucketAnalyticsConfiguration](#) REST-API oder eine gleichwertige API aus den AWS CLI oder AWS SDKs.

Wie verwende ich die Speicherklassen-Analyse?

Sie verwenden die Speicherklassen-Analyse, um Datenzugriffsmuster im Laufe der Zeit zu analysieren, um Informationen zu erhalten, die Ihnen beim Lebenszyklusmanagement Ihres STANDARD_IA-Speichers helfen. Nachdem Sie einen Filter konfiguriert haben, sehen Sie die Datenanalyse basierend auf dem Filter in der Amazon-S3-Konsole über 24 bis 48 Stunden. Die Speicherklassen-Analyse beobachtet jedoch die Zugriffsmuster einer gefilterten Datenmenge für 30 Tage oder Länger, um Informationen für eine Analyse zu sammeln, bevor ein Ergebnis zurückgegeben wird. Die Analyse wird nach dem ersten Ergebnis fortgesetzt und aktualisiert das Ergebnis, wenn sich die Zugriffsmuster ändern.

Wenn Sie einen Filter zum ersten Mal konfigurieren, dauert es möglicherweise einen Moment, um die Daten der Amazon-S3-Konsole zu analysieren.

Die Speicherklassen-Analyse beobachtet die Zugriffsmuster einer gefilterten Objekt-Datenmenge für 30 Tage oder Länger, um ausreichend viele Informationen für eine Analyse zu sammeln. Nachdem die Speicherklassen-Analyse genügend Informationen gesammelt hat, wird in der Amazon-S3-Konsole eine Meldung angezeigt, dass die Analyse abgeschlossen ist.

Bei der Durchführung der Analyse für Objekte mit seltenem Zugriff betrachtet die Speicherklassen-Analyse die gefilterte gruppierte Objektmenge basierend auf der Dauer für die sie in Amazon S3 hochgeladen waren. Die Speicherklassen-Analyse stellt fest, ob für die Altersgruppe selten ein Zugriff stattfindet, indem sie die folgenden Faktoren für die gefilterte Datenmenge betrachtet.

- Objekte in der Speicherklasse STANDARD, die größer als 128 KB sind.

- Wie viel durchschnittlichen Gesamtspeicher Sie pro Altersgruppe haben.
- Die durchschnittliche Anzahl an Bytes, die pro Altersgruppe nach außen übertragen werden (nicht die Frequenz).
- Analyseexportdaten enthalten nur Anfragen mit Daten, die für die Speicherklassen-Analyse relevant sind. Dies kann Differenzen in der Anzahl der Anfragen sowie im Hinblick auf die gesamten Bytes für Upload und Anfrage im Vergleich zu den Angaben in der Speichermetrik oder in den Ergebnissen Ihrer eigenen internen Systeme verursachen.
- Fehlgeschlagene GET- und PUT-Anfragen werden bei der Analyse nicht berücksichtigt. In den Speichermetriken sehen Sie jedoch fehlgeschlagene Anfragen.

Wie viel von meinem Speicher habe ich abgerufen?

Die Amazon-S3-Konsole zeigt in einer Grafik an, wie viel von dem Speicher in der gefilterten Datenmenge für den Beobachtungszeitraum abgerufen wurde.

Welchen Prozentsatz meines Speichers habe ich abgerufen?

Die Amazon-S3-Konsole zeigt in einer Grafik an, welcher Prozentsatz des Speichers in der gefilterten Datenmenge für den Beobachtungszeitraum abgerufen wurde.

Wie in diesem Thema bereits dargelegt, betrachtet die Speicherklassen-Analyse bei der Analyse für Objekte mit seltenem Zugriff die gefilterte Objektemenge, gruppiert nach dem Zeitpunkt, zu dem die Objekte in Amazon S3 hochgeladen wurden. Die Speicherklassen-Analyse verwendet die folgenden vordefinierten Objektaltersgruppen:

- Amazon-S3-Objekte, die weniger als 15 Tage alt sind
- Amazon S3 Objekte 15-29 Tage alt
- Amazon S3 Objekte 30-44 Tage alt
- Amazon S3 Objekte 45-59 Tage alt
- Amazon S3 Objekte 60-74 Tage alt
- Amazon S3 Objekte 75-89 Tage alt
- Amazon S3 Objekte 90-119 Tage alt
- Amazon S3 Objekte 120-149 Tage alt
- Amazon S3 Objekte 150-179 Tage alt
- Amazon S3 Objekte 180-364 Tage alt

- Amazon S3 Objekte 365-729 Tage alt
- Amazon-S3-Objekte, 730 Tage alt und älter

In der Regel benötigt sie über 30 Tage beobachtete Zugriffsmuster, um ausreichend viele Informationen für ein Analyseergebnis zu sammeln. Abhängig von dem spezifischen Zugriffsmuster Ihrer Daten könnte dies länger als 30 Tage dauern. Nachdem Sie jedoch einen Filter konfiguriert haben, sehen Sie die Datenanalyse basierend auf dem Filter in der Amazon-S3-Konsole über 24 bis 48 Stunden. Sie sehen die Analyse zum Objektzugriff auf täglicher Basis, unterteilt nach Objektaltersgruppe in der Amazon-S3-Konsole.

Wie viel von meinem Speicher wird selten abgerufen?

Die Amazon-S3-Konsole zeigt die Zugriffsmuster, die nach den vordefinierten Altersgruppen des Objekts gruppiert sind. Der angezeigte Text Frequently accessed (Häufig aufgerufen) oder Infrequently accessed (Selten aufgerufen) ist als visuelle Hilfe bei der Erstellung des Lebenszyklus gedacht.

Wie kann ich die Daten der Speicherklassen-Analyse exportieren?

Sie können festlegen, dass die Speicherklassen-Analyse Analyseberichte in einer flachen .csv-Datei (durch Kommas getrennte Werte) exportiert. Berichte werden täglich aktualisiert und basieren auf den von Ihnen konfigurierten Objektaltersgruppenfiltern. Bei Verwendung der Amazon-S3-Konsole können Sie die Exportberichtsoption wählen, wenn Sie einen Filter erstellen. Wenn Sie Dateien zum Export auswählen, geben Sie einen Ziel-Bucket und ein optionales Ziel-Präfix an, wohin die Datei geschrieben werden soll. Sie können die Daten in einen Ziel-Bucket in einem anderen Konto exportieren. Der Ziel-Bucket muss sich in derselben Region befinden wie der Bucket, dessen Analyse Sie konfiguriert haben.

Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket erstellen, um Amazon S3 Berechtigungen zu erteilen, um zu überprüfen, AWS-Konto wem der Bucket gehört, und um Objekte in den Bucket am definierten Speicherort zu schreiben. Eine Beispielrichtlinie finden Sie unter [Gewähren von Berechtigungen für S3 Inventory und S3 Analytics](#).

Nachdem Sie Speicherklassen-Analyseberichte konfiguriert haben, erhalten Sie den exportierten Bericht nach 24 Stunden täglich. Anschließend setzt Amazon S3 die Überwachung fort und stellt tägliche Berichte bereit.

Sie können die CSV-Datei in einer Tabellenkalkulationsanwendung öffnen oder die Datei in andere Anwendungen wie [Amazon QuickSight](#) importieren. Informationen zur Verwendung von Amazon S3

S3-Dateien mit Amazon QuickSight finden Sie unter [Erstellen eines Datensatzes mithilfe von Amazon S3 S3-Dateien](#) im QuickSight Amazon-Benutzerhandbuch.

Daten in den exportierten Dateien werden nach dem Datum innerhalb der Objektaltersgruppe sortiert, wie in den folgenden Beispielen gezeigt. Wenn die Speicherklasse STANDARD ist, enthält die Zeile Daten für die Spalten `ObjectAgeForSIATransition` und `RecommendedObjectAgeForSIATransition`.

Date	ConfigId	Filter	StorageClass	ObjectAge	ObjectCount	DataUploaded_MB	Storage_MB	DataRetrieved_MB	GetRequestCount	CumulativeAccessRatio	ObjectAgeForSIATransition	RecommendedObjectAgeForSIATransition
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/2/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/5/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		

Am Ende des Berichts wird die Objektaltersgruppe als ALL angegeben. Die ALL-Zeilen enthalten kumulative Summen, einschließlich Objekte, die kleiner als 128 KB sind, für alle Altersgruppen für diesen Tag.

8/24/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
9/3/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02426125	015-029	
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03545875	015-029	
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.0209529	015-029	
9/4/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02304819	015-029	
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03073092	015-029	
8/20/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014	

Der nächste Abschnitt beschreibt die im Bericht verwendeten Spalten.

Layout der exportierten Datei

In der folgenden Tabelle wird das Layout der exportierten beschrieben.

Konfigurieren der Speicherklassen-Analyse

Unter Verwendung des Tools für die Speicherklassen-Analyse von Amazon S3 können Sie Speicherzugriffsmuster analysieren, anhand derer Sie entscheiden können, ob Sie die richtigen Daten in die richtige Speicherklasse einordnen. Die Speicherklassen-Analyse beobachtet Datenzugriffsmuster analysieren, anhand derer Sie entscheiden können, wann Sie STANDARD-Speicher mit weniger häufigem Zugriff in die Speicherklasse STANDARD_IA (IA für "infrequent access" (seltener Zugriff)) überführen sollen. Weitere Informationen zu STANDARD_IA finden Sie in den [Amazon-S3-FAQ](#) und [Verwenden von Amazon-S3-Speicherklassen](#).

Sie können die Speicherklassen-Analyse Einrichten, indem Sie konfigurieren, welche Objektdaten analysiert werden sollen. Sie können die Speicherklassenanalyse für die folgenden Aufgaben konfigurieren:

- Analyse des gesamten Inhalts eines Buckets.

Sie erhalten eine Analyse für alle Objekte im Bucket.

- Analyse von nach Präfix und Tags gruppierten Objekten.

Sie können Filter konfigurieren, um Objekte nach einem gemeinsamen Präfix oder nach Objekt-Markierungen zu gruppieren, oder nach einer Kombination aus Präfix und Markierungen.

Sie erhalten für jeden konfigurierten Filter eine separate Analyse. Sie können maximal 1000 Filterkonfigurationen pro Bucket verwenden.

- Export von Analysedaten.

Wenn Sie die Speicherklassen-Analyse für einen Bucket oder Filter konfigurieren, können Sie die Analysedaten täglich in eine Datei exportieren lassen. Die Analyse für den Tag wird der Datei hinzugefügt, um ein Verlaufsanalyseprotokoll für den konfigurierten Filter zu erstellen. Die Datei wird täglich an dem Ziel Ihrer Wahl aktualisiert. Wenn Sie Dateien zum Export auswählen, geben Sie einen Ziel-Bucket und ein optionales Ziel-Präfix an, wohin die Datei geschrieben werden soll.

Sie können die Amazon S3 S3-Konsole, die REST-API AWS CLI oder die AWS SDKs verwenden, um die Speicherklassenanalyse zu konfigurieren.

Wichtig

Die Speicherklassen-Analyse gibt keine Empfehlungen für Übergänge in die Speicherklassen ONEZONE_IA oder S3 Glacier Flexible Retrieval.

Wenn Sie die Speicherklassenanalyse so konfigurieren möchten, dass Ihre Ergebnisse als CSV-Datei exportiert werden und der Ziel-Bucket die Standard-Bucket-Verschlüsselung mit `a` verwendet AWS KMS key, müssen Sie die AWS KMS Schlüsselrichtlinie aktualisieren, um Amazon S3 die Berechtigung zum Verschlüsseln der `.csv`-Datei zu erteilen. Detaillierte Anweisungen finden Sie unter [Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung](#).

Weitere Informationen zu Analytics finden Sie unter [Amazon S3 analytics – Speicherklassen-Analyse](#).

Verwenden der S3-Konsole

Konfigurieren der Speicherklassen-Analyse

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie die Speicherklassen-Analyse konfigurieren wollen.
3. Wählen Sie den Tab Metrics.
4. Wählen Sie unter Storage Class Analysis (Speicherklassen-Analyse) die Option Create analytics configuration (Analysenkonfiguration erstellen).
5. Geben Sie einen Namen für den Filter ein. Wenn der komplette Bucket analysiert werden soll, tragen Sie nichts in das Feld Prefix (Präfix) ein.
6. Geben Sie im Feld Prefix (Präfix) Text für das Präfix für die Objekte ein, die Sie analysieren möchten.
7. Um einen Tag hinzuzufügen, wählen Sie Add tag (Tag hinzufügen). Geben Sie einen Schlüssel und Wert für den Tag ein. Sie können ein Präfix und mehrere Markierungen eingeben.
8. Optional können Sie Enable (Aktivieren) unter Export CSV (CSV exportieren) auswählen, um Analyseberichte in einer flachen .csv-Datei (durch Kommas getrennte Werte) zu exportieren. Wählen Sie einen Ziel-Bucket aus, in dem die Datei gespeichert werden kann. Sie können ein Präfix für den Ziel-Bucket eingeben. Der Ziel-Bucket muss sich in demselben Bucket befinden AWS-Region wie der Bucket, für den Sie die Analyse einrichten. Der Ziel-Bucket kann sich in einem anderem AWS-Konto befinden.

Wenn der Ziel-Bucket für die .csv-Datei die Standard-Bucket-Verschlüsselung mit einem KMS-Schlüssel verwendet, müssen Sie die AWS KMS Schlüsselrichtlinie aktualisieren, um Amazon S3 die Erlaubnis zu erteilen, die .csv-Datei zu verschlüsseln. Detaillierte Anweisungen finden Sie unter [Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung](#).

9. Wählen Sie Create configuration (Konfiguration erstellen).

Amazon S3 erstellt eine Bucket-Richtlinie für den Ziel-Bucket, mit der Amazon S3 Schreibberechtigung gewährt wird. Dadurch können die Exportdaten in den Bucket geschrieben werden.

Wenn beim Erstellen der Bucket-Richtlinie ein Fehler auftritt, erhalten Sie Anweisungen zur Fehlerbehebung. Wenn Sie beispielsweise einen Ziel-Bucket in einem anderen AWS-Konto ausgewählt haben und nicht über die Berechtigungen verfügen, die Bucket-Richtlinie zu lesen und zu verändern, wird die folgende Meldung angezeigt. Sie müssen den Eigentümer des Ziel-Buckets auffordern, dem Ziel-Bucket die angezeigte Bucket-Richtlinie hinzuzufügen. Wird die Richtlinie dem Ziel-Bucket nicht hinzugefügt, erhalten Sie keine Exportdaten, weil Amazon S3 nicht die Berechtigung besitzt, in den Ziel-Bucket zu schreiben. Wenn der Quell-Bucket nicht dem Konto des aktuellen Benutzers gehört, muss die richtige Konto-ID des Quell-Buckets in der Richtlinie ersetzt werden.

Informationen über die exportierten Daten und die Funktionsweise des Filters finden Sie unter [Amazon S3 analytics – Speicherklassen-Analyse](#).

Verwenden der REST-API

Um Storage Class Analysis mithilfe der REST-API zu konfigurieren, verwenden Sie die [PutBucketAnalyticsConfiguration](#). Sie können die entsprechende Operation auch mit den AWS CLI oder AWS SDKs verwenden.

Sie können die folgenden REST-APIs verwenden, um mit Storage Class Analysis zu arbeiten:

- [DELETE Bucket Analytics configuration](#)
- [GET Bucket Analytics configuration](#)
- [List Bucket Analytics Configuration](#)

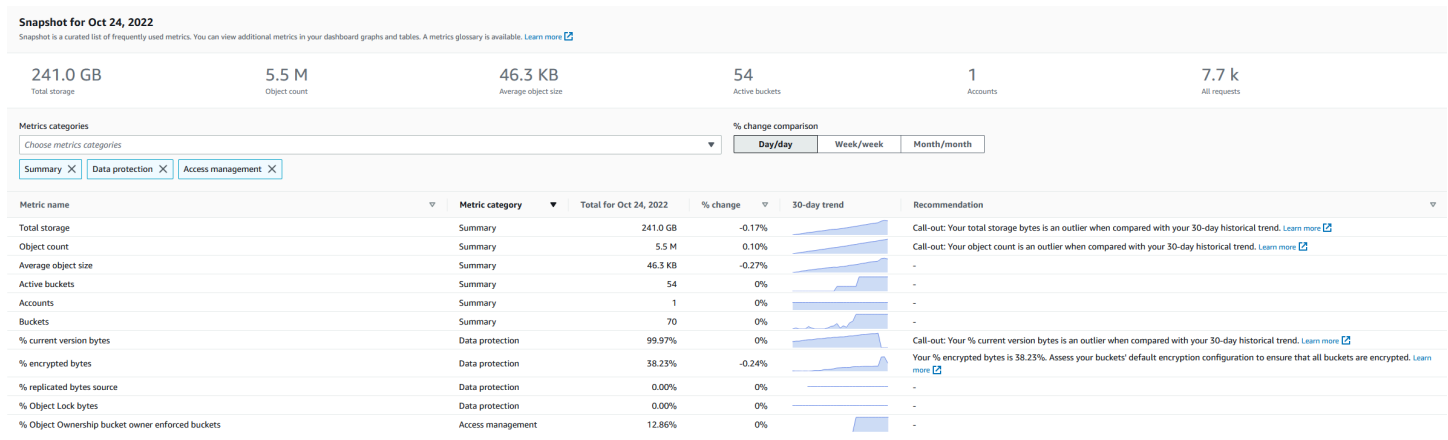
Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Erkenntnisse zu Objektspeichern und Aktivitäten erhalten können. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

Mithilfe der Metriken von S3 Storage Lens lässt sich eine Zusammenfassung der Erkenntnisse generieren. Beispielsweise können Sie herausfinden, wie viel Speicher Sie in der gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz und die Zugriffsverwaltung implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets

identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, abgebrochen werden. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport im CSV oder Parquet -Format an einen S3-Bucket senden.



Metriken und Funktionen von S3 Storage Lens

S3 Storage Lens bietet ein interaktives Standard-Dashboard, das täglich aktualisiert wird. Gemäß Vorkonfiguration von S3 Storage Lens werden in diesem Dashboard die zusammengefassten Erkenntnisse und Trends Ihres gesamten Kontos visualisiert und täglich in der S3-Konsole aktualisiert. Metriken aus diesem Dashboard werden auch in Ihrem Konto-Snapshot auf der Seite Buckets zusammengefasst. Weitere Informationen finden Sie unter [Standard-Dashboard](#).

Um weitere Dashboards zu erstellen und deren Umfang auf S3-Buckets oder Konten (für AWS Organizations) zu beschränken, erstellen Sie eine S3 Storage Lens-Dashboard-Konfiguration. AWS-Regionen Sie können S3 Storage Lens-Dashboard-Konfigurationen mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 erstellen und verwalten RESTAPI. AWS SDKs Wenn Sie ein S3-Storage-Lens-Dashboard erstellen oder bearbeiten, definieren Sie den Dashboard-Umfang und die Metrikauswahl.

S3 Storage Lens bietet neben kostenlosen Metriken auch erweiterte Metriken und Empfehlungen, auf die Sie gegen eine zusätzliche Gebühr upgraden können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen, um Erkenntnisse über Ihren Speicher zu erhalten. Zu diesen Funktionen gehören erweiterte Metrikkategorien, Präfixaggregation, kontextuelle Empfehlungen und Amazon-Veröffentlichungen. CloudWatch Präfixaggregation und kontextbezogene Empfehlungen sind nur in der Amazon-S3-Konsole verfügbar. Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter [Amazon S3-Preise](#).

Metrikkategorien

In den kostenlosen und erweiterten Stufen sind die Metriken in Kategorien unterteilt, die auf wichtige Anwendungsfälle wie Kostenoptimierung und Datenschutz abgestimmt sind. Zu den kostenlosen Metriken gehören Zusammenfassung, Kostenoptimierung, Datenschutz, Zugriffsverwaltung, Leistung und Ereignismetriken. Wenn Sie auf erweiterte Metriken und Empfehlungen upgraden, können Sie erweiterte Metriken zur Kostenoptimierung und zum Datenschutz aktivieren. Anhand dieser erweiterten Metriken können Sie Ihre S3-Speicherkosten weiter senken und Ihre Datenschutzpolitik verbessern. Sie können auch Aktivitätsmetriken und detaillierte Statuscode-Metriken aktivieren, um die Leistung von Anwendungs-Workloads zu verbessern, die auf Ihre S3-Buckets zugreifen. Weitere Informationen zu kostenlosen und erweiterten Metriken finden Sie unter [Metrikauswahl](#).

Sie können Ihren Speicher auf der Grundlage bewährter Methoden von S3 bewerten, z. B. indem Sie den Prozentsatz der Buckets analysieren, für die die Verschlüsselung, S3 Object Lock oder S3 Versioning aktiviert ist. Sie können auch potenzielle Kosteneinsparungsmöglichkeiten identifizieren. Zum Beispiel können Sie Metriken zur Anzahl von S3-Lebenszyklusregeln verwenden, um Buckets zu identifizieren, in denen Lebenszyklusregeln zum Ablauf und Übergang fehlen. Sie können auch Ihre Anforderungsaktivität pro Bucket analysieren, um Buckets zu finden, in denen Objekte in eine kostengünstigere Speicherklasse überführt werden können. Weitere Informationen finden Sie unter [Anwendungsfälle für Metriken von Amazon S3 Storage Lens](#).

Metrik-Export

Sie können nicht nur das Dashboard auf der S3-Konsole anzeigen, sondern auch Metriken in einen S3-Bucket exportieren CSV oder in einen S3-Bucket Parquet formatieren, um sie mit dem Analysetool Ihrer Wahl weiter zu analysieren. Weitere Informationen finden Sie unter [Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport](#).

CloudWatch Amazon-Verlagswesen

Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch [Dashboards](#) zu erstellen. Sie können auch CloudWatch Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch API Betriebsabläufe Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3 Storage Lens-Metriken. Die CloudWatch Veröffentlichungsoption ist für Dashboards verfügbar, die auf erweiterte Messwerte und Empfehlungen von S3 Storage Lens aktualisiert wurden. Weitere Informationen zur Unterstützung von S3 Storage Lens-Metriken finden Sie CloudWatch unter [Überwachen von Metriken von S3 Storage Lens in CloudWatch](#).

Weitere Informationen zur Verwendung von S3 Storage Lens finden Sie in den folgenden Themen.

Themen

- [Grundlegendes zu Amazon S3 Storage Lens](#)
- [Verwenden von Amazon S3 Storage Lens mit AWS Organizations](#)
- [Berechtigungen für Amazon S3 Storage Lens](#)
- [Anzeigen von Metriken mit Amazon S3 Storage Lens](#)
- [Anwendungsfälle für Metriken von Amazon S3 Storage Lens](#)
- [Amazon S3-Storage-Lens-Metrik glossar](#)
- [Arbeiten mit Amazon S3 Storage Lens unter Verwendung der Konsole und der API](#)
- [Arbeiten mit S3-Storage-Lens-Gruppen](#)

Grundlegendes zu Amazon S3 Storage Lens

Important

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon

S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar.
Weitere Informationen finden Sie unter [Standardverschlüsselung FAQ](#).

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können Metriken von S3 Storage Lens verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport im CSV oder Parquet -Format an einen S3-Bucket senden. Sie können S3 Storage Lens-Dashboards mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 REST API erstellen AWS SDKs und verwalten.

Konzepte und Terminologie von S3 Storage Lens

Dieser Abschnitt enthält die Terminologie und Konzepte, die für das Verständnis und die erfolgreiche Verwendung von Amazon S3 Storage Lens unerlässlich sind.

Themen

- [Konfiguration des Dashboards](#)
- [Standard-Dashboard](#)

- [Dashboards](#)
- [Konto-Snapshot](#)
- [Metrik-Export](#)
- [Heimatregion](#)
- [Aufbewahrungszeitraum](#)
- [Metrikkategorien](#)
- [Empfehlungen](#)
- [Metrikauswahl](#)
- [S3 Storage Lens und AWS Organizations](#)

Konfiguration des Dashboards

S3 Storage Lens erfordert eine Dashboard-Konfiguration, die die Eigenschaften enthält, mit denen Metriken in Ihrem Namen für ein einzelnes Dashboard oder einen Export aggregiert werden. Wenn Sie eine Konfiguration erstellen, wählen Sie den Dashboard-Namen und die Heimatregion aus, die Sie nach der Erstellung des Dashboards nicht mehr ändern können. Sie können optional Tags hinzufügen und einen Metrikexport im CSV oder Parquet -Format konfigurieren.

In der Dashboard-Konfiguration definieren Sie auch den Dashboard-Umfang und die Metrikauswahl. Der Umfang kann den gesamten Speicherplatz für Ihr Organisationskonto oder Abschnitte umfassen, die nach Region, Bucket und Konto gefiltert sind. Wenn Sie die Metrikauswahl konfigurieren, wählen Sie zwischen kostenlosen Metriken und erweiterten Metriken und Empfehlungen, auf die Sie gegen eine zusätzliche Gebühr upgraden können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen. Zu diesen Funktionen gehören erweiterte Metrikkategorien, Aggregation auf Präfixebene, kontextuelle Empfehlungen und Amazon-Veröffentlichungen. CloudWatch Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter [Amazon S3-Preise](#).

Standard-Dashboard

Das Standard-Dashboard von S3 Storage Lens auf der Konsole hat den Namen `default-account-dashboard`. Gemäß S3-Vorkonfiguration werden in diesem Dashboard die zusammengefassten Erkenntnisse und Trends Ihres gesamten Kontos visualisiert und täglich in der S3-Konsole aktualisiert. Sie können den Konfigurationsbereich des Standard-Dashboards zwar nicht ändern, haben aber die Möglichkeit, ein Upgrade für die Metrikauswahl durchzuführen und statt der kostenlosen Metriken die erweiterten Metriken und Empfehlungen zu verwenden. Sie können den

optionalen Metrikexport konfigurieren oder sogar das Dashboard deaktivieren. Löschen können Sie das Standard-Dashboard jedoch nicht.

Note

Wenn Sie Ihr Standard-Dashboard deaktivieren, wird es nicht mehr aktualisiert. Sie erhalten auf Ihrem S3 Storage Lens-Dashboard, in Ihrem Metrikexport oder im Konto-Snapshot auf der S3-Seite Buckets keine neuen täglichen Metriken mehr. Wenn Ihr Dashboard erweiterte Metriken und Empfehlungen verwendet, werden Ihnen keine Gebühren mehr berechnet. Sie können im Dashboard weiterhin historische Daten anzeigen, bis der Zeitraum von 14 Tagen für Datenabfragen abläuft. Wenn Sie erweiterte Metriken und Empfehlungen aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf historische Daten zuzugreifen.

Dashboards

Sie können zusätzliche S3 Storage Lens-Dashboards erstellen und deren Umfang nach AWS-Regionen S3-Buckets oder Konten (für AWS Organizations) festlegen. Wenn Sie ein S3-Storage-Lens-Dashboard erstellen oder bearbeiten, definieren Sie den Dashboard-Umfang und die Metrikauswahl. S3 Storage Lens bietet neben kostenlosen Metriken auch erweiterte Metriken und Empfehlungen, auf die Sie gegen eine zusätzliche Gebühr upgraden können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen, um Erkenntnisse über Ihren Speicher zu erhalten. Dazu gehören erweiterte Metrikkategorien, Aggregation auf Präfixebene, kontextuelle Empfehlungen und Amazon-Veröffentlichungen. CloudWatch Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter [Amazon S3-Preise](#).

Sie können das Dashboard auch deaktivieren oder löschen. Wenn Sie ein Dashboard deaktivieren, wird es nicht mehr aktualisiert und Sie erhalten keine neuen täglichen Metriken mehr. Sie können historische Daten dann noch bis zum Ende des Ablaufzeitraums von 14 Tagen einsehen. Wenn Sie erweiterte Metriken und Empfehlungen für das Dashboard aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf historische Daten zuzugreifen.

Wenn Sie ein Dashboard löschen, gehen alle Dashboard-Konfigurationseinstellungen verloren. Sie erhalten dann keine neuen täglichen Metriken mehr und können nicht mehr auf die mit dem Dashboard verbundenen historischen Daten zugreifen. Wenn Sie auf die historischen Daten eines

gelöschten Dashboards zugreifen möchten, müssen Sie ein neues Dashboard mit demselben Namen in derselben Heimatregion erstellen.

Note

- Sie können S3 Storage Lens verwenden, um bis zu 50 Dashboards pro Heimatregion zu erstellen.
- Dashboards auf Organisationsebene können nur auf einen regionalen Bereich beschränkt werden.

Konto-Snapshot

Der Konto-Snapshot von S3 Storage Lens fasst Metriken aus Ihrem Standard-Dashboard zusammen und zeigt Ihren Gesamtspeicher, die Objektzahl und die durchschnittliche Objektgröße auf der Seite Buckets der S3-Konsole an. Mit diesem Konto-Snapshot haben Sie schnellen Zugriff auf Erkenntnisse über Ihren Speicher, ohne die Seite Buckets verlassen zu müssen. Der Konto-Snapshot bietet auch mit einem Klick Zugriff auf Ihr interaktives S3-Storage-Lens-Dashboard.

Sie können Ihr Dashboard verwenden, um Erkenntnisse und Trends zu visualisieren, Ausreißer zu kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden einzusehen. Ihr Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, Bucket-, Objekt- oder sogar Präfixebene zu generieren. Sie können auch einmal täglich einen Metrikexport im Oder-Format an einen S3-Bucket senden. CSV Parquet

Sie können den Dashboard-Umfang des default-account dashboard (Standardkonto-Dashboards) nicht ändern, da er mit dem Account snapshot (Konto-Snapshot) verknüpft ist. Sie können die Metrikauswahl in Ihrem jedoch default-account-dashboarvon kostenlosen Metriken auf kostenpflichtige erweiterte Metriken und Empfehlungen umstellen. Nach dem Upgrade können Sie alle Anfragen, hochgeladenen Bytes und heruntergeladenen Bytes im Account snapshot (Konto-Snapshot) von S3 Storage Lens anzeigen.

Note

Wenn Sie Ihr Standard-Dashboard deaktivieren, wird Ihr Account snapshot (Konto-Snapshot) nicht mehr aktualisiert. Um weiterhin Metriken im Konto-Snapshot anzuzeigen, können Sie den default-account-dashboardeerneut aktivieren.

Metrik-Export

Ein Metrik-Export ist eine Datei, die alle in Ihrer S3-Storage-Lens-Konfiguration erfassten Metriken enthält. Diese Informationen werden täglich im CSV Parquet OR-Format generiert und an einen S3-Bucket gesendet. Sie können den Metrikexport für weitere Analysen verwenden, indem Sie das Metrik-Tool Ihrer Wahl verwenden. Der S3-Bucket für den Export Ihrer Metriken muss sich in derselben Region wie Ihre S3-Storage-Lens-Konfiguration befinden. Sie können einen Metrikexport von S3 Storage Lens über die S3-Konsole generieren, indem Sie Ihre Dashboard-Konfiguration bearbeiten. Sie können einen Metrikexport auch mithilfe von AWS CLI und konfigurieren AWS SDKs.

Heimatregion

In der Heimatregion werden alle AWS-Region S3 Storage Lens-Metriken für eine bestimmte Dashboard-Konfiguration gespeichert. Sie müssen eine Heimatregion auswählen, wenn Sie Ihr S3-Storage-Lens-Dashboard konfigurieren. Nachdem Sie eine Heimatregion ausgewählt haben, können Sie sie nicht mehr ändern. Wenn Sie eine Storage-Lens-Gruppe erstellen, empfehlen wir außerdem, dieselbe Heimatregion wie für das Storage-Lens-Dashboard auszuwählen.

Note

Sie können eine der folgenden Regionen als Heimatregion auswählen:

- US East (N. Virginia) – us-east-1
- US East (Ohio) – us-east-2
- US West (N. California) – us-west-1
- US West (Oregon) – us-west-2
- Asia Pacific (Mumbai) – ap-south-1
- Asien-Pazifik (Seoul) – ap-northeast-2
- Asia Pacific (Singapore) – ap-southeast-1
- Asien-Pazifik (Sydney) – ap-southeast-2
- Asien-Pazifik (Tokio) – ap-northeast-1
- Canada (Central) – ca-central-1
- China (Peking) – cn-north-1
- China (Ningxia) – cn-northwest-1
- Europe (Frankfurt) – eu-central-1
- Europe (Irland) – eu-west-1

- Europe (London) – eu-west-2
- Europe (Paris) – eu-west-3
- Europe (Stockholm) – eu-north-1
- South America (São Paulo) – sa-east-1

Aufbewahrungszeitraum

S3-Storage-Lens-Metriken werden beibehalten, damit Sie historische Trends erkennen und Unterschiede bei Speichernutzung und -aktivität im Zeitverlauf vergleichen können. Sie können Metriken von Amazon S3 Storage Lens für Abfragen verwenden, um historische Trends anzuzeigen und Unterschiede in Ihrer Speichernutzung und -aktivität im Zeitverlauf zu vergleichen.

Alle Metriken für S3 Storage Lens werden für einen Zeitraum von 15 Monaten aufbewahrt. Metriken sind jedoch nur für Abfragen für eine bestimmte Dauer verfügbar, die von Ihrer [Metrikauswahl](#) abhängt. Diese Dauer kann nicht geändert werden. Kostenlose Metriken stehen für Abfragen für einen Zeitraum von 14 Tagen zur Verfügung, und fortschrittliche Metriken stehen für Abfragen für 15 Monate zur Verfügung.

Metrikkategorien

In den kostenlosen und erweiterten Stufen sind die Metriken von S3 Storage Lens in Kategorien unterteilt, die auf wichtige Anwendungsfälle wie Kostenoptimierung und Datenschutz abgestimmt sind. Zu den kostenlosen Metriken gehören Zusammenfassung, Kostenoptimierung, Datenschutz, Zugriffsverwaltung, Leistung und Ereignismetriken. Wenn Sie auf erweiterte Metriken und Empfehlungen upgraden, können Sie zusätzliche Metriken zur Kostenoptimierung und zum Datenschutz aktivieren, mit denen Sie Ihre S3-Speicherkosten weiter senken und den Datenschutz gewährleisten können. Sie können auch Aktivitätsmetriken und detaillierte Statuscode-Metriken aktivieren, um die Leistung von Anwendungs-Workloads zu verbessern.

Die folgende Liste enthält alle kostenlosen und erweiterten Metrikkategorien. Eine vollständige Liste der einzelnen Metriken, die in jeder Kategorie enthalten sind, finden Sie im [Metrik-Glossar](#).

Zusammenfassende Metriken

Zusammenfassende Metriken bieten allgemeine Erkenntnisse über Ihren S3-Speicher, einschließlich Ihrer gesamten Speicherbytes und der Anzahl der Objekte.

Metriken zur Kostenoptimierung

Metriken zur Kostenoptimierung liefern Erkenntnisse, die Sie zur Verwaltung und Optimierung Ihrer Speicherkosten nutzen können. Sie können beispielsweise Buckets identifizieren, unvollständige mehrteilige Uploads enthalten, die älter als 7 Tage sind.

Mit erweiterten Metriken und Empfehlungen können Sie erweiterte Metriken zur Kostenoptimierung aktivieren. Zu diesen Metriken gehören Metriken für die Anzahl der S3-Lebenszyklusregeln, mit denen Sie die Anzahl der S3-Lebenszyklusregeln pro Bucket für den Ablauf und für den Übergang ermitteln können.

Metriken zum Datenschutz

Metriken zum Datenschutz liefern Erkenntnisse über Datenschutzfunktionen wie Verschlüsselung und S3 Versioning. Sie können diese Metriken verwenden, um Buckets zu identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen. Sie können beispielsweise Buckets identifizieren, die keine Standardverschlüsselung mit AWS Key Management Service Schlüsseln (SSE-KMS) oder S3-Versionierung verwenden.

Mit erweiterten Metriken und Empfehlungen können Sie erweiterte Metriken zum Datenschutz aktivieren. Zu diesen Metriken gehören Metriken für die Anzahl der Replikationsregeln pro Bucket.

Metriken zur Zugriffsverwaltung

Metriken zur Zugriffsverwaltung liefern Erkenntnisse im Hinblick auf S3 Object Ownership. Sie können diese Metriken verwenden, um zu sehen, welche Einstellungen Ihre Buckets für die Objekteigentümerschaft verwenden.

Ereignismetriken

Ereignismetriken bieten Einblicke in S3-Ereignisbenachrichtigungen. Mithilfe von Ereignismetriken können Sie sehen, für welche Buckets S3-Ereignisbenachrichtigungen konfiguriert sind.

Leistungsmetriken

Leistungsmetriken liefern relevante Erkenntnisse für S3 Transfer Acceleration. Anhand von Leistungsmetriken können Sie sehen, für welche Buckets Transfer Acceleration aktiviert ist.

Aktivitätsmetriken (erweitert)

Wenn Sie das Dashboard auf erweiterte Metriken und Empfehlungen upgraden, können Sie Aktivitätsmetriken aktivieren. Aktivitätsmetriken liefern Details darüber, wie Ihr Speicherplatz angefordert wird (z. B. All-Anforderungen, Get-Anforderungen, Put-Anforderungen), sowie über hoch- oder heruntergeladene Bytes und Fehler.

Anhand von Aktivitätsmetriken auf Präfixebene können Sie ermitteln, welche Präfixe selten verwendet werden, sodass Sie [im S3-Lebenszyklus zu einer optimaleren Speicherklasse wechseln](#) können.

Detaillierte Statuscode-Metriken (erweitert)

Wenn Sie das Dashboard auf erweiterte Metriken und Empfehlungen upgraden, können Sie detaillierte Statuscode-Metriken aktivieren. Detaillierte Statuscode-Metriken bieten Einblicke in HTTP Statuscodes wie 403 Forbidden und 503 Service Unavailable, die Sie zur Behebung von Zugriffs- oder Leistungsproblemen verwenden können. Sie können sich beispielsweise die Metrik 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) ansehen, um Workloads zu identifizieren, die auf Buckets zugreifen, ohne dass die korrekten Berechtigungen angewendet wurden.

Detaillierte Statuscode-Metriken auf Präfixebene können verwendet werden, um ein besseres Verständnis der Vorkommen von HTTP Statuscodes anhand von Präfixen zu erhalten. Mit den Metriken zur Anzahl der 503-Fehler können Sie beispielsweise Präfixe ermitteln, die während der Datenerfassung Drosselungsanfragen erhalten.

Empfehlungen

S3 Storage Lens bietet automatisierte Empfehlungen, die Sie bei der Optimierung Ihres Speichers unterstützen. Die Empfehlungen werden kontextbasiert neben relevanten Metriken im S3-Storage-Lens-Dashboard platziert. Historische Daten kommen nicht für Empfehlungen in Betracht, da Empfehlungen nur für aktuelle Geschehnisse relevant sind. Empfehlungen werden nur angezeigt, wenn sie relevant sind.

Es gibt folgende Arten von S3-Storage-Lens-Empfehlungen:

- Vorschläge

Vorschläge weisen Sie auf Trends beim Speicher und bei Aktivitäten hin, die auf Möglichkeiten zur Optimierung der Speicherkosten oder auf bewährte Datenschutzmethoden hindeuten könnten. Anhand der vorgeschlagenen Themen können Sie im Amazon-S3-Benutzerhandbuch und im S3-Storage-Lens-Dashboard nach weiteren Informationen zu den spezifischen Regionen, Buckets oder Präfixen suchen.

- Callouts

Callouts sind Empfehlungen, die Sie auf interessante Anomalien in Bezug auf den Speicher und Aktivitäten aufmerksam machen, die über einen bestimmten Zeitraum auftreten und möglicherweise Maßnahmen erfordern oder überwacht werden sollten.

- Ausreißer-Callouts

S3 Storage Lens bietet Callouts für Metriken, bei denen es sich auf der Basis Ihres Trends der letzten 30 Tage um Ausreißer handelt. Der Ausreißer wird mit einem Standardwert berechnet, der auch als Z-Score bezeichnet wird. Bei diesem Score wird die Metrik des aktuellen Tages vom Durchschnitt der letzten 30 Tage für die Metrik subtrahiert. Die Metrik des aktuellen Tages wird dann durch die Standardabweichung für die Metrik in den letzten 30 Tagen dividiert. Der resultierende Score liegt normalerweise zwischen -3 und +3. Diese Zahl stellt die Anzahl der Standardabweichungen dar, die die Metrik des aktuellen Tages vom Mittelwert entfernt ist.

Metriken mit einem Score von > 2 oder < -2 werden in S3 Storage Lens als Ausreißer erachtet, da sie über oder unter dem Wert von 95 Prozent der normal verteilten Daten liegen.

- Callouts für signifikante Änderungen

Das Callout für signifikante Änderungen gilt für Metriken, die sich erwartungsgemäß selten ändern. Daher ist er auf eine höhere Sensitivität eingestellt als die Ausreißerberechnung, die sich typischerweise im Bereich von ± 20 Prozent gegenüber dem Vortag, der Vorwoche oder dem Vormonat befindet.

Bearbeiten von Callouts für den Speicher und Aktivitäten – Wenn Sie ein Callout für signifikante Änderungen erhalten, muss das nicht auf ein Problem hinweisen. Das Callout könnte das Ergebnis einer erwarteten Änderung des Speichers sein. Beispielsweise können Sie vor Kurzem eine große Anzahl neuer Objekte hinzugefügt, eine große Anzahl von Objekten gelöscht oder ähnliche geplante Änderungen vorgenommen haben.

Wenn Sie in Ihrem Dashboard ein Callout für signifikante Änderungen sehen, sollten Sie zunächst bestimmen, ob es durch die jüngsten Umstände erklärt werden kann. Ist dies nicht der Fall, können Sie über das S3-Storage-Lens-Dashboard weitere Details abrufen, um herauszufinden, welchen spezifischen Regionen, Buckets oder Präfixen die Fluktuation zugrunde liegt.

- Erinnerungen

Erinnerungen bieten Einblicke in die Funktionsweise von Amazon S3. Sie können Ihnen helfen, mehr darüber zu erfahren, wie Sie S3-Funktionen nutzen können, um die Speicherkosten zu senken oder bewährte Datenschutzmethoden anzuwenden.

Metrikauswahl

S3 Storage Lens bietet zwei Arten von Metriken, die Sie für das Dashboard und den Export auswählen können: kostenlose Metriken und fortschrittliche Metriken und Empfehlungen.

- **Kostenlose Metriken**

S3 Storage Lens bietet kostenlose Metriken für alle Dashboards und Konfigurationen. Kostenlose Metriken enthalten relevante Metriken für Ihre Speichernutzung, z. B. die Anzahl der Buckets und der Objekte in Ihrem Konto. Zu den kostenlosen Metriken gehören auch auf Anwendungsfällen basierende Metriken (z. B. Metriken zur Kostenoptimierung und zum Datenschutz), anhand derer Sie untersuchen können, ob Ihr Speicher gemäß den bewährten Methoden von S3 konfiguriert ist. Alle kostenlosen Metriken werden täglich gesammelt. Daten stehen 14 Tage für Abfragen zur Verfügung. Weitere Informationen darüber, welche Metriken kostenlos sind, finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

- **Fortschrittliche Metriken und Empfehlungen**

S3 Storage Lens bietet kostenlose Metriken für alle Dashboards und Konfigurationen mit der Option zum Upgrade auf fortschrittliche Metriken und Empfehlungen. Es fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Zu den erweiterten Metriken und Empfehlungen gehören neben den kostenlosen Metriken weitere Metriken, wie erweiterte Metriken zum Datenschutz und zur Kostenoptimierung, Aktivitätsmetriken und detaillierte Statuscode-Metriken. Erweiterte Metriken und Empfehlungen umfassen auch Empfehlungen zur Optimierung Ihrer Speichernutzung. Die Empfehlungen werden kontextbasiert neben relevanten Metriken im Dashboard platziert.

Erweiterte Metriken und Empfehlungen umfassen die folgenden Funktionen:

- **Erweiterte Metriken** – Generieren Sie zusätzliche Metriken. Eine vollständige Liste der erweiterten Metrikkategorien finden Sie unter [Metrikkategorien](#). Eine vollständige Liste der Metriken finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).
- **Amazon CloudWatch Publishing** — Veröffentlicht S3 Storage Lens-Metriken CloudWatch , um einen einheitlichen Überblick über Ihren Betriebsstatus in CloudWatch [Dashboards](#) zu erhalten. Sie können auch CloudWatch API Betriebsabläufe und Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Weitere Informationen finden Sie unter [Überwachen von Metriken von S3 Storage Lens in CloudWatch](#).

- Präfixaggregation – Metriken werden auf Präfixebene gesammelt. Durch die Aktivierung der Präfixaggregation werden alle in der Dashboard-Konfiguration enthaltenen Metriken auf Präfixebene erweitert. Metriken werden nur für Präfixe generiert, die den konfigurierten Schwellenwert erreichen. Beachten Sie, dass Metriken, die auf Präfixebene gelten, mit Präfixaggregation verfügbar sind. Ausgenommen sind Einstellungen auf Bucket-Ebene und Metriken zur Regelanzahl. Metriken auf Präfix-Ebene werden nicht veröffentlicht in CloudWatch
- Aggregation von Storage-Lens-Gruppen – Metriken werden auf Storage-Lens-Gruppenebene gesammelt. Nachdem Sie Erweiterte Metriken und Empfehlungen und Aggregation von Storage-Lens-Gruppen aktiviert haben, können Sie angeben, welche Storage-Lens-Gruppen in das Storage-Lens-Dashboard aufgenommen oder daraus ausgeschlossen werden sollen. Es muss mindestens eine Storage-Lens-Gruppe angegeben werden. Die angegebenen Storage-Lens-Gruppen müssen sich in der angegebenen Heimatregion des Dashboard-Kontos befinden. Metriken auf Gruppenebene von Storage Lens werden nicht veröffentlicht. CloudWatch

Alle fortschrittlichen Metriken werden täglich gesammelt. Daten stehen für Abfragen 15 Monate lang zur Verfügung. Weitere Informationen zu den Speichermetriken, die von S3 Storage Lens aggregiert werden, finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Note

Empfehlungen sind nur über das S3-Storage-Lens-Dashboard in der Amazon-S3-Konsole verfügbar.

S3 Storage Lens und AWS Organizations

AWS Organizations ist ein Tool AWS-Service, mit dem Sie alle Ihre Organisationshierarchie AWS-Konten unter einer einzigen Organisation zusammenfassen können. Amazon S3 Storage Lens bietet AWS Organizations eine zentrale Ansicht des Objektspeichers und der Aktivitäten in Ihrem Amazon S3 Speicher.

Weitere Informationen finden Sie unter [Verwenden von Amazon S3 Storage Lens mit AWS Organizations](#).

- Vertrauenswürdiger Zugriff

Sie müssen den vertrauenswürdigen Zugriff im Verwaltungskonto Ihrer Organisation aktivieren, damit S3 Storage Lens Speichermetriken und Nutzungsdaten für alle Mitgliedskonten in Ihrer

Organisation aggregieren kann. Dann können Sie über das Verwaltungskonto Dashboards oder Exporte für Ihre Organisation erstellen oder delegierten Administratoren Zugriff auf andere Konten in der Organisation gewähren.

Sie können den vertrauenswürdigen Zugriff für S3 Storage Lens jederzeit deaktivieren, um die Aggregation von Metriken für Ihre Organisation zu beenden.

- Delegated Administrator

Sie können Dashboards und Metriken für S3 Storage Lens für Ihre Organisation erstellen, indem Sie Ihr AWS Organizations Verwaltungskonto verwenden oder indem Sie delegierten Administratorzugriff auf andere Konten in Ihrer Organisation gewähren. Sie können delegierte Administratoren jederzeit abmelden. Durch Abmelden eines delegierten Administrators wird die Aggregation neuer Speichermetriken durch alle Dashboards auf Organisationsebene, die von diesem delegierten Administrator erstellt wurden, automatisch beendet.

Weitere Informationen finden Sie unter [Amazon S3 Storage Lens und AWS Organizations](#) im AWS Organizations -Benutzerhandbuch.

Serviceverknüpfte Rollen für Amazon S3 Storage Lens

Neben dem AWS Organizations vertrauenswürdigen Zugriff verwendet Amazon S3 Storage Lens AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger IAM Rollentyp, der direkt mit S3 Storage Lens verknüpft ist. Serviceverknüpfte Rollen werden von S3 Storage Lens vordefiniert und enthalten alle Berechtigungen, die zum Erfassen der täglichen Speicher- und Aktivitätsmetriken aus Mitgliedskonten in Ihrer Organisation benötigt werden.

Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens](#).

Verwenden von Amazon S3 Storage Lens mit AWS Organizations

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können Metriken von S3 Storage Lens verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für

den Datenschutz implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

Sie können Amazon S3 Storage Lens verwenden, um Speichermetriken und Nutzungsdaten für alle zu sammeln AWS-Konten, die Teil Ihrer AWS Organizations Hierarchie sind. Dazu müssen Sie den vertrauenswürdigen Zugriff von AWS Organizations S3 Storage Lens mithilfe Ihres AWS Organizations Verwaltungskontos verwenden und aktivieren.

Nachdem Sie den vertrauenswürdigen Zugriff aktiviert haben, können Sie anderen Konten in Ihrer Organisation den delegierten Administratorzugriff gewähren. Diese Konten können dann S3-Storage-Lens-Konfigurationen und -Dashboards erstellen, die unternehmensweite Speichermetriken und Benutzerdaten sammeln.

Weitere Informationen zum Aktivieren des vertrauenswürdigen Zugriffs finden Sie unter [Amazon S3 Storage Lens und AWS Organizations](#) im AWS Organizations -Benutzerhandbuch.

Themen

- [Aktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens](#)
- [Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens](#)
- [Registrieren eines delegierten Administrators für S3 Storage Lens](#)
- [Aufheben der Registrierung eines delegierten Administrators für S3 Storage Lens](#)

Aktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens

Durch die Aktivierung des vertrauenswürdigen Zugriffs gewähren Sie Amazon S3 Storage Lens den Zugriff auf Ihre AWS Organizations Hierarchie, Mitgliedschaft und Struktur im Rahmen von AWS Organizations API Vorgängen. S3 Storage Lens wird dann zu einem vertrauenswürdigen Service für die Struktur Ihrer gesamten Organisation.

Immer wenn eine Dashboard-Konfiguration definiert wird, erstellt S3 Storage Lens serviceverknüpfte Rollen in den Verwaltungskonten oder den delegierten Administratorkonten Ihrer Organisation. Die serviceverknüpfte Rolle erteilt S3 Storage Lens die Berechtigung für folgende Aufgaben:

- Beschreiben von Organisationen
- Auflisten von Konten
- Überprüfen Sie eine AWS-Service Zugriffsliste für die Organisationen
- Abrufen delegierter Administratoren für die Organisationen

S3 Storage Lens kann dann sicherstellen, dass es Zugriff hat, um die kontoübergreifende Metriken für die Konten in Ihren Organisationen zu sammeln. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens](#).

Nachdem Sie den vertrauenswürdigen Zugriff aktiviert haben, können Sie anderen Konten in Ihrer Organisation den delegierten Administratorzugriff zuweisen. Wenn ein Konto als delegierter Administrator für einen Dienst gekennzeichnet ist, erhält das Konto die Autorisierung für den Zugriff auf alle schreibgeschützten Organisationsvorgänge. API Dieser Zugriff macht die Mitglieder und Strukturen Ihrer Organisation für den delegierten Administrator sichtbar, sodass dieser ebenfalls S3-Storage-Lens-Dashboards erstellen können.

Note

Der vertrauenswürdige Zugriff für Amazon S3 Storage Lens kann nur mit dem Verwaltungskonto aktiviert werden.

Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens

Durch die Deaktivierung des vertrauenswürdigen Zugriffs beschränken Sie die S3-Storage-Lens-Funktionalität auf die Kontoebene. Darüber hinaus sieht jeder Kontoinhaber nur die Informationen von S3 Storage Lens für den Umfang seines Kontos und nicht für die gesamte Organisation. Alle Dashboards, die einen vertrauenswürdigen Zugriff erfordern, werden nicht mehr aktualisiert, behalten jedoch ihre historischen Daten für den Zeitraum bei, in dem [Daten für Abfragen verfügbar sind](#).

Note

- Durch Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens wird außerdem die Erfassung und Aggregation von Speichermetriken durch alle Dashboards auf Organisationsebene automatisch beendet.

- Ihre Verwaltungs- und delegierten Administratorkonten können während des Zeitraums, in dem die Daten für Abfragen verfügbar sind, weiterhin die historischen Daten für Ihre bestehenden Dashboards auf Organisationsebene anzeigen.

Registrieren eines delegierten Administrators für S3 Storage Lens

Dashboards auf Organisationsebene können mithilfe des Verwaltungskontos Ihrer Organisation oder mit delegierten Administratorkonten erstellt werden. Der delegierte Administratorzugriff ermöglicht anderen Konten neben dem Verwaltungskonto, Dashboards auf Organisationsebene zu erstellen. Das Registrieren und Aufheben der Registrierung anderer Konten als delegierte Administratoren für eine Organisation kann lediglich über das Verwaltungskonto erfolgen.

Informationen zum Registrieren eines delegierten Administrators über die Amazon-S3-Konsole finden Sie unter [Registrieren von delegierten Administratoren für S3 Storage Lens](#).

Sie können einen delegierten Administrator auch registrieren, indem Sie das AWS Organizations REST API AWS CLI, oder SDKs vom Verwaltungskonto aus verwenden. Weitere Informationen finden Sie [RegisterDelegatedAdministrator](#) in der AWS Organizations APIReferenz.

Note

Bevor Sie mithilfe von, oder SDKs einen delegierten Administrator bestimmen können AWS Organizations REST API AWS CLI, müssen Sie den [EnableAWSOrganizationsAccess](#)Vorgang aufrufen.

Aufheben der Registrierung eines delegierten Administrators für S3 Storage Lens

Sie können eine delegiertes Administratorkontos auch abmelden. Der delegierte Administratorzugriff ermöglicht anderen Konten neben dem Verwaltungskonto, Dashboards auf Organisationsebene zu erstellen. Konten, die als delegierte Administratoren festgelegt wurden, können nur über das Verwaltungskonto einer Organisation abgemeldet werden.

Informationen zum Abmelden eines delegierten Administrators über die S3-Konsole finden Sie unter [Aufheben der Registrierung delegierter Administratoren für S3 Storage Lens](#).

Sie können einen delegierten Administrator auch mithilfe von AWS Organizations RESTAPI, AWS CLI oder SDKs vom Verwaltungskonto abmelden. Weitere Informationen finden Sie [DeregisterDelegatedAdministrator](#) in der Referenz.AWS Organizations API

Note

- Durch Abmelden eines delegierten Administrators wird die Aggregation neuer Speichermetriken durch alle Dashboards auf Organisationsebene, die von diesem delegierten Administrator erstellt wurden, automatisch beendet.
- Die abgemeldeten delegierten Administratoren können weiterhin die historischen Daten für die Dashboards, die sie erstellt haben, anzeigen, während Daten für Abfragen verfügbar sind.

Berechtigungen für Amazon S3 Storage Lens

Amazon S3 Storage Lens benötigt neue Berechtigungen in AWS Identity and Access Management (IAM), um den Zugriff auf S3 Storage Lens-Aktionen zu autorisieren. Um diese Berechtigungen zu gewähren, können Sie eine IAM identitätsbasierte Richtlinie verwenden. Sie können diese Richtlinie IAM Benutzern, Gruppen oder Rollen zuordnen, um ihnen Berechtigungen zu erteilen. Diese Berechtigungen können die Fähigkeit beinhalten, S3 Storage Lens zu aktivieren oder zu deaktivieren oder auf ein beliebiges Dashboard oder eine Konfiguration von S3 Storage Lens zuzugreifen.

Der IAM Benutzer oder die Rolle muss zu dem Konto gehören, das das Dashboard oder die Konfiguration erstellt hat oder besitzt, es sei denn, die beiden folgenden Bedingungen sind erfüllt:

- Ihr Konto ist Mitglied von AWS Organizations.
- Sie haben über Ihr Verwaltungskonto als delegierter Administrator Zugriff auf die Erstellung von Dashboards auf Organisationsebene erhalten.

Note

- Sie können Dashboards von Amazon S3 Storage Lens nicht mithilfe der Root-Benutzer-Anmeldeinformationen Ihres Kontos aufrufen. Um auf die S3 Storage Lens-Dashboards zugreifen zu können, müssen Sie einem neuen oder vorhandenen IAM Benutzer die erforderlichen IAM Berechtigungen erteilen. Anschließend können Sie sich mit diesen

Anmeldeinformationen anmelden, um auf S3-Storage-Lens-Dashboards zuzugreifen. Weitere Informationen finden Sie unter [Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch](#).

- Die Verwendung von S3 Storage Lens in der Amazon S3-Konsole kann mehrere Berechtigungen erfordern. Zum Bearbeiten eines Dashboards in der Konsole benötigen Sie beispielsweise die folgenden Berechtigungen:
 - `s3:ListStorageLensConfigurations`
 - `s3:GetStorageLensConfiguration`
 - `s3:PutStorageLensConfiguration`

Themen

- [Festlegen von Kontoberechtigungen für die Verwendung von S3 Storage Lens](#)
- [Festlegen von Kontoberechtigungen für die Verwendung von S3-Storage-Lens-Gruppen](#)
- [Festlegen von Berechtigungen für die Verwendung von S3 Storage Lens mit AWS Organizations](#)

Festlegen von Kontoberechtigungen für die Verwendung von S3 Storage Lens

Zum Erstellen und Verwalten von S3-Storage-Lens-Dashboards und Storage-Lens-Dashboard-Konfigurationen benötigen Sie je nachdem, welche Aktionen Sie ausführen möchten, die folgenden Berechtigungen:

IAMBerechtigungen im Zusammenhang mit Amazon S3 Storage Lens

Aktion	IAMBerechtigungen
Erstellen oder Aktualisieren eines S3-Storage-Lens-Dashboards in der Amazon S3-Konsole.	<code>s3:ListStorageLensConfigurations</code> <code>s3:GetStorageLensConfiguration</code> <code>s3:GetStorageLensConfigurat ionTagging</code> <code>s3:PutStorageLensConfiguration</code> <code>s3:PutStorageLensConfigurat ionTagging</code>

Aktion	IAMBerechtigungen
Abrufen der Tags eines S3-Storage-Lens-Dashboards in der Amazon-S3-Konsole.	s3:ListStorageLensConfigurations s3:GetStorageLensConfigurati onTagging
Ansehen eines S3-Storage-Lens-Dashboards in der Amazon S3-Konsole.	s3:ListStorageLensConfigurations s3:GetStorageLensConfiguration s3:GetStorageLensDashboard
Löschen eines S3-Storage-Lens-Dashboards in der Amazon S3-Konsole.	s3:ListStorageLensConfigurations s3:GetStorageLensConfiguration s3>DeleteStorageLensConfigu ration
Erstellen oder aktualisieren Sie eine S3-Storage-Lens-Konfiguration mit dem AWS CLI oder einem AWS SDK.	s3:PutStorageLensConfiguration s3:PutStorageLensConfigurat ionTagging
Rufen Sie die Tags einer S3 Storage Lens-Konfiguration ab, indem Sie AWS CLI oder ein verwenden AWS SDK.	s3:GetStorageLensConfigurat ionTagging
Zeigen Sie eine S3-Storage-Lens-Konfiguration an, indem Sie AWS CLI oder ein verwenden AWS SDK.	s3:GetStorageLensConfiguration
Löschen Sie eine S3-Storage-Lens-Konfiguration mithilfe von AWS CLI oder AWS SDK.	s3>DeleteStorageLensConfigu ration

Note

- Die Dashboard-Ansichten von S3 Storage Lens werden CloudTrail mit dem Namen des Ereignisses angemeldet `GetStorageLensDashboardDataInternal`.
- Sie können Ressourcen-Tags in einer IAM Richtlinie verwenden, um Berechtigungen zu verwalten.
- Ein IAM Benutzer oder eine Rolle mit diesen Berechtigungen kann Metriken aus Buckets und Präfixen sehen, für die er möglicherweise keine direkte Berechtigung hat, Objekte zu lesen oder aufzulisten.
- Wenn bei S3-Storage-Lens-Dashboards mit aktivierten Metriken auf Präfixebene ein ausgewählter Präfixpfad mit einem Objektschlüssel übereinstimmt, zeigt das Dashboard den Objektschlüssel möglicherweise als anderes Präfix an.
- Für Metrikexporte, die in einem Bucket in Ihrem Konto gespeichert sind, werden Berechtigungen mithilfe der in der IAM Richtlinie vorhandenen `s3:GetObject` Berechtigungen erteilt. Ähnlich können bei einer AWS Organizations Entität das Verwaltungskonto der Organisation oder delegierte Administratorkonten IAM Richtlinien verwenden, um Zugriffsberechtigungen für Dashboards und Konfigurationen auf Organisationsebene zu verwalten.

Festlegen von Kontoberechtigungen für die Verwendung von S3-Storage-Lens-Gruppen

Mithilfe von S3-Storage-Lens-Gruppen können Sie die Speicherverteilung innerhalb von Buckets anhand von Präfix, Suffix, Objekt-Tag, Objektgröße oder Objektalter nachvollziehen. Sie können Storage-Lens-Gruppe an Dashboards anhängen, um ihre aggregierten Metriken anzuzeigen.

Für die Verwendung von Storage-Lens-Gruppen benötigen Sie bestimmte Berechtigungen. Weitere Informationen finden Sie unter [the section called “Berechtigungen für Storage-Lens-Gruppen”](#).

Festlegen von Berechtigungen für die Verwendung von S3 Storage Lens mit AWS Organizations

Sie können Amazon S3 Storage Lens verwenden, um Speichermetriken und Nutzungsdaten für alle Konten zu sammeln, die Teil Ihrer AWS Organizations Hierarchie sind. Im Folgenden finden Sie die Aktionen und Berechtigungen für die Verwendung von S3 Storage Lens mit Organizations.

AWS Organizations verwandte IAM Berechtigungen für die Verwendung von S3 Storage Lens

Aktion	IAMBerechtigungen
Aktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens für Ihre Organisation.	<code>organizations:EnableAWSServiceAccess</code>
Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens für Ihre Organisation.	<code>organizations:DisableAWSServiceAccess</code>
Registrieren eines delegierten Administrators für die Erstellung von S3-Storage-Lens-Dashboards oder -Konfigurationen für Ihre Organisation.	<code>organizations:RegisterDelegatedAdministrator</code>
Abmelden eines delegierten Administrators, damit dieser keine S3 Storage Lens-Dashboards oder -Konfigurationen mehr für Ihre Organisation erstellen kann.	<code>organizations:DeregisterDelegatedAdministrator</code>
Zusätzliche Berechtigungen zum Erstellen von organisationsweiten S3-Storage-Lens-Konfigurationen	<code>organizations:DescribeOrganization</code> <code>organizations:ListAccounts</code> <code>organizations:ListAWSServiceAccessForOrganization</code> <code>organizations:ListDelegatedAdministrators</code> <code>iam:CreateServiceLinkedRole</code>

Anzeigen von Metriken mit Amazon S3 Storage Lens

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte

Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden.

Standardmäßig sind alle Dashboards mit kostenlosen Metriken konfiguriert. Dazu gehören Metriken, anhand derer Sie die Nutzung und Aktivität in Ihrem S3-Speicher nachvollziehen, Ihre Speicherkosten optimieren und bewährte Methoden für Datenschutz und Zugriffsverwaltung implementieren können. Kostenlose Metriken werden bis auf Bucket-Ebene aggregiert. Bei kostenlosen Metriken stehen Daten bis zu 14 Tage lang für Abfragen zur Verfügung.

Zu den erweiterten Metriken und Empfehlungen gehören die folgenden zusätzlichen Funktionen, mit denen Sie weitere Einblicke in die Nutzung und Aktivität in Ihrem Speicher sowie bewährte Methoden zur Speicheroptimierung erhalten können:

- Kontextbezogene Empfehlungen (nur im Dashboard verfügbar)
- Erweiterte Metriken (einschließlich nach Buckets aggregierter Aktivitätsmetriken)
- Präfixzusammenfassung
- Aggregation von Storage-Lens-Gruppen
- Aggregation von Storage-Lens-Gruppen
- Amazon-CloudWatch-Veröffentlichung

Erweiterte Metrikdaten stehen 15 Monate für Abfragen zur Verfügung. Für die Verwendung von S3 Storage Lens mit fortschrittlichen Metriken fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#). Weitere Informationen über kostenlose und erweiterte Metriken finden Sie unter [Metrikauswahl](#).

Themen

- [Anzeigen von S3-Storage-Lens-Metriken in den Dashboards](#)
- [Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport](#)
- [Überwachen von Metriken von S3 Storage Lens in CloudWatch](#)

Anzeigen von S3-Storage-Lens-Metriken in den Dashboards

In der Amazon-S3-Konsole bietet S3 Storage Lens ein interaktives Standard-Dashboard, mit dem Sie Erkenntnisse zu Daten sowie entsprechende Trends visualisieren können. Zudem lässt sich

das Dashboard verwenden, um Ausreißer zu kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten sowie zur Anwendung bewährter Datenschutzmethoden einzusehen. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Konto-, Bucket-, AWS-Region-, Präfix- oder Storage-Lens-Gruppenebene zu generieren. Wenn Sie S3 Storage Lens für AWS Organizations aktiviert haben, besteht zudem die Möglichkeit, Erkenntnisse auf Organisationsebene zu generieren (z. B. Daten für alle Konten, die der AWS Organizations-Hierarchie angehören). Das Dashboard wird immer für das letzte Datum geladen, für das Metriken verfügbar sind.

Das Standard-Dashboard von S3 Storage Lens in der Konsole heißt `default-account-dashboard`. Gemäß Amazon-S3-Vorkonfiguration werden in diesem Dashboard die zusammengefassten Erkenntnisse und Trends Ihres gesamten Kontos visualisiert und täglich in der S3-Konsole aktualisiert. Sie können den Konfigurationsbereich des Standard-Dashboards nicht ändern, aber Sie haben die Möglichkeit, statt der kostenlosen Metriken die kostenpflichtigen erweiterten Metriken und Empfehlungen auszuwählen. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen. Zu diesen Funktionen gehören erweiterte Metrikkategorien, Aggregation auf Präfixebene, kontextbezogene Empfehlungen und Amazon-CloudWatch-Veröffentlichung.

Sie können das Standard-Dashboard deaktivieren, aber nicht löschen. Wenn Sie Ihr Standard-Dashboard deaktivieren, wird es nicht mehr aktualisiert. Sie erhalten keine neuen täglichen Metriken mehr in S3 Storage Lens oder im Konto-Snapshot auf der Seite Buckets. Sie können im Standard-Dashboard weiterhin historische Daten anzeigen, bis der Zeitraum von 14 Tagen für Datenabfragen abläuft. Wenn Sie erweiterte Metriken und Empfehlungen aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Standard-Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf diese Daten zuzugreifen.

Sie können zusätzliche S3-Storage-Lens-Dashboards erstellen und ihren Umfang nach AWS-Regionen, S3-Buckets oder Konten festlegen. Zudem sind Sie in der Lage, Dashboards auch nach Organisation einzugrenzen, wenn Sie Storage Lens für AWS Organizations aktiviert haben. Wenn Sie ein S3-Storage-Lens-Dashboard erstellen oder bearbeiten, definieren Sie den Dashboard-Umfang und die Metrikauswahl.

Sie können zusätzliche Dashboards, die Sie erstellen, deaktivieren oder löschen.

- Wenn Sie ein Dashboard deaktivieren, wird es nicht mehr aktualisiert und Sie erhalten keine neuen täglichen Metriken mehr. Sie können historische Daten für kostenlose Metriken dann noch bis zum Ende des Ablaufzeitraums von 14 Tagen einsehen. Wenn Sie erweiterte Metriken und

Empfehlungen für das Dashboard aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf diese Daten zuzugreifen.

- Wenn Sie ein Dashboard löschen, gehen alle Dashboard-Konfigurationseinstellungen verloren. Sie erhalten dann keine neuen täglichen Metriken mehr und können nicht mehr auf die mit dem Dashboard verbundenen historischen Daten zugreifen. Wenn Sie auf die historischen Daten eines gelöschten Dashboards zugreifen möchten, müssen Sie ein neues Dashboard mit demselben Namen in derselben Heimatregion erstellen.

Themen

- [Anzeigen eines Amazon S3-Storage-Lens-Dashboards](#)
- [Grundlegendes zum S3-Storage-Lens-Dashboard](#)

Anzeigen eines Amazon S3-Storage-Lens-Dashboards

Das folgende Verfahren zeigt, wie Sie ein S3-Storage-Lens-Dashboard in der S3-Konsole ansehen können. Anwendungsfallbasierte Anleitungen, die zeigen, wie Sie das Dashboard verwenden können, um Ihre Kosten zu optimieren, bewährte Verfahren zu implementieren und die Leistung von Anwendungen zu verbessern, die auf S3-Buckets zugreifen, finden Sie unter [Anwendungsfälle für Metriken von Amazon S3 Storage Lens](#).


Note

Sie können Dashboards von Amazon S3 Storage Lens nicht mithilfe der Root-Benutzer-Anmeldeinformationen Ihres Kontos aufrufen. Für den Zugriff auf S3-Storage-Lens-Dashboards müssen Sie die erforderlichen AWS Identity and Access Management (IAM)-Berechtigungen einem neuen oder bestehenden IAM-Benutzer erteilen. Anschließend können Sie sich mit diesen Anmeldeinformationen anmelden, um auf S3-Storage-Lens-Dashboards zuzugreifen. Weitere Informationen finden Sie unter [Berechtigungen für Amazon S3 Storage Lens](#) und [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.

Ihr Dashboard wird in S3 Storage Lens geöffnet. Der Abschnitt Snapshot for date (Snapshot für Datum) zeigt das letzte Datum an, an dem S3 Storage Lens Metriken gesammelt hat. Ihr Dashboard wird immer für das letzte Datum geladen, für das Metriken verfügbar sind.

4. (Optional) Wenn Sie das Datum für Ihr S3-Storage-Lens-Dashboard ändern möchten, wählen Sie in der oberen rechten Datumsauswahl ein neues Datum aus.
5. (Optional) Gehen Sie wie folgt vor, wenn Sie temporäre Filter anwenden möchten, mit denen Sie den Umfang Ihrer Dashboard-Daten weiter einschränken können:
 - a. Erweitern Sie den Bereich Filter.
 - b. Wenn Sie nach bestimmten Konten, AWS-Regionen, Speicherklassen, Buckets, Präfixen oder Storage-Lens-Gruppen filtern möchten, wählen Sie die Optionen aus, nach denen gefiltert werden soll.

 Note

Der Filter Präfixe und der Filter Storage-Lens-Gruppen können nicht gleichzeitig angewendet werden.

- c. Wenn Sie einen Filter aktualisieren möchten, wählen Sie Apply (Anwenden) aus.
 - d. Zum Entfernen eines Filters klicken Sie auf das X neben dem Filter.
6. Wählen Sie zum Anzeigen von Daten für eine bestimmte Metrik in einem beliebigen Abschnitt Ihres S3-Storage-Lens-Dashboards im Feld Metric (Metrik) den Namen der entsprechenden Metrik aus.
7. In jedem Diagramm oder jeder Visualisierung des S3-Storage-Lens-Dashboards können Sie mithilfe der Registerkarten Konto, AWS-Regionen, Speicherklassen, Buckets, Präfixe oder Storage-Lens-Gruppen tiefere Aggregationsebenen aufrufen. Ein Beispiel finden Sie unter [Entdecken Sie kalte Amazon S3-Buckets](#).

Grundlegendes zum S3-Storage-Lens-Dashboard

Ihr S3-Storage-Lens-Dashboard enthält eine primäre Registerkarte Overview (Übersicht) und bis zu fünf zusätzlichen Registerkarten für die einzelnen Aggregationsebenen:

- Konten
- AWS-Regionen

- Speicherklassen
- Buckets
- Präfixe
- Storage-Lens-Gruppen

Auf der Registerkarte Overview (Übersicht) werden Ihre Dashboard-Daten in drei verschiedene Abschnitte zusammengefasst: Snapshot for date (Snapshot für Datum), Trends and distributions (Trends und Verteilungen) und Top N overview (Top N-Übersicht).

Weitere Informationen zur Verwendung Ihres S3-Storage-Lens-Dashboards finden Sie in den folgenden Abschnitten.

Snapshot

Der Abschnitt Snapshot for date (Snapshot für Datum) zeigt zusammenfassende Metriken an, die S3 Storage Lens für das ausgewählte Datum gesammelt hat. Diese zusammenfassenden Metriken umfassen die folgenden Metriken:

- Gesamtspeicher – Die Gesamtmenge des verwendeten Speichers in Byte
- Anzahl der Objekte – Die Gesamtanzahl der Objekte im AWS-Konto
- Durchschnittliche Objektgröße – Die durchschnittliche Objektgröße
- Aktive Buckets – Die Gesamtanzahl der aktiven Buckets in aktiver Nutzung mit Speicher > 0 Byte im Konto
- Konten – Die Anzahl der Konten, deren Speicher zum Umfang gehören. Dieser Wert ist 1, es sei denn, Sie verwenden AWS Organizations und S3 Storage Lens hat vertrauenswürdigen Zugriff mit einer gültigen serviceverknüpften Rolle. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens](#).
- Buckets – Die Gesamtanzahl der Buckets im Konto

Metrik-Daten

Für jede Metrik, die im Snapshot angezeigt wird, können Sie die folgenden Daten sehen:

- Metrikname – Der Name der Metrik
- Metrikkategorie – Die Kategorie, in der die Metrik organisiert ist
- Gesamtsumme für Datum – Die Gesamtzahl für das ausgewählte Datum

- % Änderung – Die prozentuale Änderung seit dem letzten Snapshot-Datum
- 30-Tage-Trend – Eine Trendlinie, die die Änderungen der Metrik über einen Zeitraum von 30 Tagen zeigt
- Empfehlung – Eine kontextbezogene Empfehlung, die auf den Daten basiert, die im Snapshot bereitgestellt werden. Empfehlungen sind mit erweiterten Metriken und Empfehlungen verfügbar. Weitere Informationen finden Sie unter [Empfehlungen](#).

Metrikkategorien

Sie können optional Ihren Dashboard-Abschnitt Snapshot for date (Snapshot für Datum) aktualisieren, um Metriken für andere Kategorien anzuzeigen. Wenn Sie Snapshot-Daten für zusätzliche Metriken sehen möchten, können Sie aus den folgenden Metrics categories (Metrikkategorien) wählen:

- Kostenoptimierung
- Datenschutz
- Aktivität (mit erweiterten Metriken verfügbar)
- Zugriffsverwaltung
- Leistung
- Ereignisse

Im Abschnitt Snapshot for date (Snapshot für Datum) wird nur eine Auswahl von Metriken für jede Kategorie angezeigt. Um alle Metriken für eine bestimmte Kategorie anzuzeigen, wählen Sie die Metrik in den Abschnitten Trends and distributions (Trends und Verteilungen) oder Top N overview (Top N-Übersicht) aus. Weitere Informationen zu den Metrikkategorien finden Sie unter [Metrikkategorien](#). Eine vollständige Liste der Metriken von S3 Storage Lens finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Trends und Verteilungen

Der zweite Abschnitt der Registerkarte Overview (Übersicht) lautet Trends and distributions (Trends und Verteilungen). Im Abschnitt Trends and distributions (Trends und Verteilungen) können Sie zwei Metriken auswählen, um sie über einen von Ihnen definierten Zeitraum zu vergleichen. Der Abschnitt Trends and distributions (Trends und Verteilungen) zeigt die Beziehung zwischen zwei Metriken im Zeitverlauf. Dieser Abschnitt enthält Diagramme, mit denen Sie die Verteilung von Storage class (Speicherklasse) und Region zwischen den beiden Trends sehen können, die Sie beobachten. Sie

können optional einen Datenpunkt in einem der Diagramme aufschlüsseln, um ihn eingehender zu analysieren.

Eine Anleitung, die den Abschnitt Trends and distribution (Trends und Verteilungen) verwendet, finden Sie unter [Identifizieren von Buckets, die keine serverseitige Verschlüsselung mit AWS KMS für die Standardverschlüsselung \(SSE-KMS\) verwenden](#).

Top N-Übersicht

Der dritte Abschnitt des S3-Storage-Lens-Dashboards ist die Top-N-Übersicht (in aufsteigender oder absteigender Reihenfolge sortiert). In diesem Abschnitt werden die ausgewählten Metriken in den Top-N-Konten, -AWS-Regionen, -Buckets, -Präfixen oder Storage-Lens-Gruppen angezeigt. Wenn Sie S3 Storage Lens für AWS Organizations aktiviert haben, können Sie die ausgewählten Metriken für die Organisation einsehen.

Eine Anleitung, die den Abschnitt Top N overview (Top N-Übersicht) verwendet, finden Sie unter [Identifizieren Sie Ihre größten S3-Buckets](#).

Aufschlüsselung und Analyse nach Optionen

Um für eine reibungslose Analyse zu sorgen, bietet das S3-Storage-Lens-Dashboard ein Menü mit Aktionen, das bei Auswahl eines Diagrammwerts angezeigt wird. Wenn Sie dieses Menü verwenden möchten, wählen Sie einen beliebigen Diagrammwert aus, um die zugehörigen Metrikwerte zu sehen. Es stehen zwei Optionen zur Auswahl:

- Mit der Aktion Drill down (Aufgliedern) wird der ausgewählte Wert als Filter auf alle Registerkarten Ihres Dashboards angewendet. Sie können dann einen Drilldown für eine ausführlichere Analyse durchführen.
- Bei Auswahl der Aktion Analysieren nach wird die ausgewählte Registerkarte Dimension im Dashboard geöffnet und der Wert wird als Filter angewendet. Zu diesen Registerkarten gehören Konten, AWS-Regionen, Speicherklassen, Buckets, Präfixe (für Dashboards mit aktivierten Optionen Erweiterte Metriken und Präfixaggregation) sowie Storage-Lens-Gruppen (für Dashboards mit aktivierten Optionen Erweiterte Metriken und Aggregation von Storage-Lens-Gruppen). Mit Analysieren nach lassen sich die Daten im Kontext der neuen Dimension anzeigen und noch ausführlicher analysieren.

Die Aktionen Aufgliedern und Analysieren nach sind möglicherweise deaktiviert, wenn das Ergebnis unlogische Ergebnisse liefern würde oder keinen Wert hätte. Beide Aktionen Aufgliedern und

Analysieren nach führen dazu, dass Filter auf alle Registerkarten des Dashboards angewendet werden (zusätzlich zu den vorhandenen Filtern). Bei Bedarf lassen sich Filter auch entfernen.

Registerkarten

Die Registerkarten auf Dimensionsebene bieten eine detaillierte Ansicht aller Werte innerhalb einer bestimmten Dimension. Die Registerkarte AWS-Regionen zeigt beispielsweise Metriken für alle AWS-Regionen an und die Registerkarte Buckets Metriken für alle Buckets. Jeder Dimensions-Tab hat das gleiche Layout, das aus vier Abschnitten besteht:

- Im Trenddiagramm werden Ihre Top-N-Elemente innerhalb der Dimension in den letzten 30 Tagen für die ausgewählte Metrik angezeigt. Standardmäßig enthält dieses Diagramm die Top 10 Elemente. Sie können die Anzahl auf minimal 3 Elemente verringern oder auf maximal 50 Elemente erhöhen.
- Das Histogramm enthält ein vertikales Balkendiagramm für die ausgewählte Kombination aus Datum und Metrik. Wenn Sie eine große Anzahl von Elementen in diesem Diagramm anzeigen möchten, müssen Sie ggf. horizontal scrollen.
- Im Blasen-Analysediagramm werden alle Elemente innerhalb der Dimension dargestellt. In diesem Diagramm wird die erste Metrik auf der X-Achse und die zweite Metrik auf der Y-Achse dargestellt. Die dritte Metrik wird durch die Größe der Blase repräsentiert.
- Die metrische Rasteransicht enthält jedes Element in der Dimension. Die Werte werden in Zeilen aufgeführt. Die Spalten stellen jede verfügbare Metrik dar, die zur einfacheren Navigation nach Tabs mit Metrikkategorien angeordnet sind.

Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport

Amazon S3-Storage-Lens-Metriken werden täglich als Metrik-Exportdateien im CSV- oder Apache Parquet-Format generiert und in einem S3-Bucket in Ihrem Konto abgelegt. Von dort aus können Sie den Metrikexport in die Analysetools Ihrer Wahl wie Amazon QuickSight und Amazon Athena aufnehmen, wo Sie die Speichernutzung und die Aktivitätstrends analysieren können.

Themen

- [Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln](#)
- [Was ist ein S3-Storage-Lens-Exportmanifest?](#)
- [Grundlegendes zum Amazon S3-Storage-Lens-Exportschema](#)

Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln

Um Amazon S3 Storage Lens die Berechtigung zum Verschlüsseln Ihrer Metrikexporte mit einem kundenverwalteten Schlüssel zu erteilen, müssen Sie eine Schlüsselrichtlinie verwenden. Gehen Sie folgendermaßen vor, um Ihre Schlüsselrichtlinie zu aktualisieren, damit Sie S3-Storage-Lens-Metrik-Exporte mit einem KMS-Schlüssel verschlüsseln können.

So gewähren Sie S3 Storage Lens Berechtigungen zum Verschlüsseln mithilfe Ihres KMS-Schlüssels

1. Melden Sie sich mit dem AWS Management Console vom Kunden verwalteten Schlüssel an AWS-Konto , dem der Kunde gehört.
2. Öffnen Sie die AWS KMS Konsole unter <https://console.aws.amazon.com/kms>.
3. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
4. Klicken Sie im linken Navigationsbereich auf Customer managed keys (Vom Kunden verwaltete Schlüssel).
5. Wählen Sie unter Vom Kunden verwaltete Schlüssel den Schlüssel aus, den Sie zur Verschlüsselung der Metrikexporte verwenden möchten. AWS KMS keys sind regionsspezifisch und müssen sich in derselben Region befinden wie der S3-Bucket, in dem der Metrikexport ausgeführt wird.
6. Wählen Sie unter Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
7. Um die Schlüsselrichtlinie zu aktualisieren, wählen Sie Edit (Bearbeiten).
8. Fügen Sie unter Edit key policy (Schlüsselrichtlinie bearbeiten) die folgende Schlüsselrichtlinie zu der vorhandenen Schlüsselrichtlinie hinzu. Wenn Sie diese Richtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Informationen.

```
{
  "Sid": "Allow Amazon S3 Storage Lens use of the KMS key",
  "Effect": "Allow",
  "Principal": {
    "Service": "storage-lens.s3.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
```

```
    "StringEquals": {
      "aws:SourceArn": "arn:aws:s3:us-east-1:source-account-id:storage-
lens/your-dashboard-name",
      "aws:SourceAccount": "source-account-id"
    }
  }
}
```

9. Wählen Sie Änderungen speichern aus.

Weitere Informationen zum Erstellen von kundenverwalteten Schlüsseln und zum Verwenden von Schlüsselrichtlinien finden Sie unter den folgenden Themen im AWS Key Management Service Entwicklerhandbuch:

- [Erste Schritte](#)
- [Verwenden der wichtigsten Richtlinien in AWS KMS](#)

Sie können die AWS KMS PUT Schlüsselrichtlinien-API-Operation ([PutKeyPolicy](#)) auch verwenden, um die Schlüsselrichtlinie in die vom Kunden verwalteten Schlüssel zu kopieren, die Sie zur Verschlüsselung der Metrikexporte mithilfe der REST-API und der SDKs verwenden möchten. AWS CLI

Was ist ein S3-Storage-Lens-Exportmanifest?

Angesichts der großen Menge von aggregierten Daten kann der tägliche Metrik-Export von S3 Storage Lens in mehrere Dateien aufgeteilt werden. Die Manifestdatei `manifest.json` beschreibt, wo sich die Metrik-Exportdateien für den jeweiligen Tag befinden. Immer wenn ein neuer Export bereitgestellt wird, wird ein neues zugehöriges Manifest erstellt. Jedes in der `manifest.json`-Datei enthaltene Manifest bietet Metadaten und andere grundlegende Informationen zum Export.

Die Manifestinformationen umfassen die folgenden Eigenschaften:

- `sourceAccountId` – Die Konto-ID des Konfigurationsinhabers.
- `configId` – Eine eindeutige ID für das Dashboard.
- `destinationBucket` – Der Amazon-Ressourcenname (ARN) des Ziel-Buckets, in dem der Metrik-Export abgelegt wird.
- `reportVersion` – Die Version des Exports.
- `reportDate` – Das Datum des Berichts.

- `reportFormat` – Das Format des Berichts.
- `reportSchema` – Das Schema des Berichts.
- `reportFiles` – Die tatsächliche Liste der Exportberichtsdateien, die sich im Ziel-Bucket befinden.

Nachfolgend sehen Sie ein Beispiel eines Manifests in einer `manifest.json`-Datei für einen CSV-formatierten Export.

```
{
  "sourceAccountId": "123456789012",
  "configId": "my-dashboard-configuration-id",
  "destinationBucket": "arn:aws:s3:::destination-bucket",
  "reportVersion": "V_1",
  "reportDate": "2020-11-03",
  "reportFormat": "CSV",

  "reportSchema": "version_number, configuration_id, report_date, aws_account_number, aws_region, stor
  "reportFiles": [
    {
      "key": "DestinationPrefix/StorageLens/123456789012/my-dashboard-
configuration-id/V_1/reports/dt=2020-11-03/a38f6bc4-2e3d-4355-ac8a-e2fdcf3de158.csv",
      "size": 1603959,
      "md5Checksum": "2177e775870def72b8d84febe1ad3574"
    }
  ]
}
```

Nachfolgend sehen Sie ein Beispiel eines Manifests in einer `manifest.json`-Datei für einen Export im Parquet-Format.

```
{
  "sourceAccountId": "123456789012",
  "configId": "my-dashboard-configuration-id",
  "destinationBucket": "arn:aws:s3:::destination-bucket",
  "reportVersion": "V_1",
  "reportDate": "2020-11-03",
  "reportFormat": "Parquet",
  "reportSchema": "message s3.storage.lens { required string version_number;
required string configuration_id; required string report_date; required string
aws_account_number; required string aws_region; required string storage_class;
```

```

required string record_type; required string record_value; required string
bucket_name; required string metric_name; required long metric_value; }",
  "reportFiles":[
    {
      "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-configuration-
id/V_1/reports/dt=2020-11-03/bd23de7c-b46a-4cf4-bcc5-b21aac5be0f5.par",
      "size":14714,
      "md5Checksum":"b5c741ee0251cd99b90b3e8eff50b944"
    }
  ]
}

```

Sie können Ihren Metrikexport so konfigurieren, dass er als Teil Ihrer Dashboard-Konfiguration in der Amazon S3 S3-Konsole oder mithilfe der Amazon S3 S3-REST-API und SDKs generiert wird. AWS CLI

Grundlegendes zum Amazon S3-Storage-Lens-Exportschema

Die folgende Tabelle enthält das Schema Ihres S3-Storage-Lens-Metrik-Exports.

Attributname	Datentyp	Spaltenname	Beschreibung
VersionNumber	Zeichenfolge	version_number	Die Version der verwendeten S3-Storage-Lens-Metriken.
ConfigurationId	String	configuration_id	Die configuration_id Ihrer S3-Storage-Lens-Konfiguration.
ReportDate	String	report_date	Das Datum, an dem die Metriken erfasst wurden.
AwsAccountNumber	String	aws_account_number	Ihre AWS-Konto Nummer.

Attributname	Datentyp	Spaltenname	Beschreibung
AwsRegion	String	aws_region	Die, AWS-Region für die die Metriken verfolgt werden.
StorageClass	String	storage_class	Die Speicherklasse des betreffenden Buckets.
RecordType	ENUM	record_type	Der Typ des gemeldeten Artefakts (ACCOUNT, BUCKET oder PREFIX).
RecordValue	String	record_value	Der Wert des RecordType - Artefakts. <div data-bbox="1187 976 1510 1243" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note Der record_value ist URL-kodiert.</p> </div>
BucketName	String	bucket_name	Der Name des Buckets, der gemeldet wird.
MetricName	Zeichenfolge	metric_name	Der Name der Metrik, die gemeldet wird.
MetricValue	Long	metric_value	Der Wert der Metrik, die gemeldet wird.

Beispiel für einen S3-Storage-Lens-Metrik-Export

Im Folgenden sehen Sie ein Beispiel für einen auf diesem Schema basierenden S3-Storage-Lens-Metrik-Export.

Note

Sie können Metriken für Storage-Lens-Gruppen ermitteln, indem Sie in der Spalte `record_type` nach dem Wert `STORAGE_LENS_GROUP_BUCKET` oder `STORAGE_LENS_GROUP_ACCOUNT` suchen. In der Spalte `record_value` wird der Amazon-Ressourcenname (ARN) für die Storage-Lens-Gruppe angezeigt (z. B. `arn:aws:s3:us-east-1:123456789012:storage-lens-group/slg-1`).

version	configuration_id	report_date	aws_account_number	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			StorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedStorageBytes	20000
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedObjectCount	20
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedStorageBytes	2478828742
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedObjectCount	1598961
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			DeleteMarkerObjectCount	1500
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionStorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: StorageBytes			29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: ObjectCount			12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: ReplicatedStorageBytes			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: ReplicatedObjectCount			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: EncryptedStorageBytes			29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: EncryptedObjectCount			12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: DeleteMarkerObjectCount			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: ObjectLockEnabledStorageBytes			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: ObjectLockEnabledObjectCount			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: CurrentVersionStorageBytes			29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: CurrentVersionObjectCount			12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: NonCurrentVersionStorageBytes			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: NonCurrentVersionObjectCount			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: IncompleteMultipartUploadStorageBytes			0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu: cloudtrail-log-sf: IncompleteMultipartUploadObjectCount			0

Im Folgenden finden Sie ein Beispiel für einen S3-Storage-Lens-Metrikexport mit Storage-Lens-Gruppendaten.

version_number	configuration_id	report_date	aws_account_num	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		StorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUObjectCountOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	StorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUObjectCountOlderThan7Days	0


Überwachen von Metriken von S3 Storage Lens in CloudWatch

Sie können S3-Storage-Lens-Metriken in Amazon CloudWatch veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch-[Dashboards](#) zu erstellen. Sie können auch CloudWatch-Funktionen wie Alarme und ausgelöste Aktionen, Metrikmathematik und Anomalieerkennung verwenden, um S3-Storage-Lens-Metriken zu überwachen und Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch-API-Operationen Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3-Storage-Lens-Metriken. Weitere Informationen zu CloudWatch finden Sie im [Amazon-CloudWatch-Benutzerhandbuch](#).

Sie können die CloudWatch-Veröffentlichungsoption für neue oder vorhandene Dashboard-Konfigurationen mithilfe der Amazon-S3-Konsole, der Amazon-S3-REST-API, AWS CLI und AWS SDKs aktivieren. Dashboards, die auf fortschrittliche Metriken und Empfehlungen von S3 Storage Lens aktualisiert werden, können die CloudWatch-Veröffentlichungsoption verwenden. Informationen zu Preisen für erweiterte Metriken und Empfehlungen von S3 Storage Lens finden Sie unter [Amazon-S3-Preise](#). Es fallen keine zusätzlichen Gebühren für die Veröffentlichung von CloudWatch-Metriken an. Es gelten jedoch andere CloudWatch-Gebühren z. B. für Dashboards, Alarme und API-Aufrufe. Weitere Informationen hierzu finden Sie unter [Amazon CloudWatch – Preise](#).

Die Metriken von S3 Storage Lens werden in CloudWatch in dem Konto veröffentlicht, das die S3-Storage-Lens-Konfiguration besitzt. Nachdem Sie die CloudWatch-Veröffentlichungsoption für erweiterte Metriken und Empfehlungen aktiviert haben, können Sie in CloudWatch auf Metriken auf

Organisations-, Konto- und Bucket-Ebene zugreifen. Metriken auf Präfixebene sind in CloudWatch nicht verfügbar.

 Note

S3-Storage-Lens-Metriken sind tägliche Metriken und werden einmal täglich in CloudWatch veröffentlicht. Wenn Sie S3-Storage-Lens-Metriken in CloudWatch abfragen, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3-Storage-Lens-Metriken in Ihrem S3-Storage-Lens-Dashboard in der Amazon-S3-Konsole angezeigt werden, kann es ein paar Stunden dauern, bis diese Metriken in CloudWatch angezeigt werden. Wenn Sie die CloudWatch-Veröffentlichungsoption für S3-Storage-Lens-Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken auf CloudWatch veröffentlicht werden.

Nachdem Sie die CloudWatch-Veröffentlichungsoption aktiviert haben, können Sie die folgenden CloudWatch-Funktionen verwenden, um Ihre Daten in S3 Storage Lens zu überwachen und zu analysieren:

- [Dashboards](#) – Erstellen Sie mithilfe von CloudWatch-Dashboards benutzerdefinierte Dashboards für S3 Storage Lens. Teilen Sie Ihr CloudWatch-Dashboard mit Personen, die keinen direkten Zugriff auf Ihr AWS-Konto haben, teamübergreifend, mit Stakeholdern und mit Personen außerhalb Ihrer Organisation.
- [Alarmer und ausgelöste Aktionen](#) – Konfigurieren Sie Alarmer, die Metriken überwachen und Maßnahmen ergreifen, wenn ein Schwellenwert überschritten wird. Sie können beispielsweise einen Alarm konfigurieren, der eine Amazon-SNS-Benachrichtigung sendet, wenn die Metrik Incomplete Multipart Upload Bytes (Unvollständige Bytes für mehrteilige Uploads) an drei aufeinanderfolgenden Tagen 1 GB überschreitet.
- [Anomalieerkennung](#) – Ermöglichen Sie die Anomalieerkennung, um Metriken kontinuierlich zu analysieren, normale Baselines und Oberflächenanomalien zu ermitteln. Sie können einen Anomalieerkennungsalarm basierend auf dem erwarteten Wert einer Metrik erstellen. Sie können beispielsweise Anomalien für die Metrik Object Lock Enabled Bytes (Bytes mit aktivierter Objektsperre), um das unbefugte Entfernen von Objektsperreinstellungen zu erkennen.
- Mithilfe von [Metrikberechnungen](#) können Sie mehrere -Metriken mit S3-Storage-Lens-Metriken abfragen und mathematische Ausdrücke verwenden, um neue Zeitreihen basierend auf diesen Metriken zu erstellen. Sie können beispielsweise eine neue Metrik erstellen, um die

durchschnittliche Objektgröße zu erhalten, indem Sie `StorageBytes` durch `ObjectCount` dividieren.

Weitere Informationen zur CloudWatch-Veröffentlichungsoption für S3-Storage-Lens-Metriken finden Sie in den folgenden Themen.

Themen

- [Metriken und Dimensionen von S3 Storage Lens](#)
- [Aktivieren der CloudWatch-Veröffentlichung für S3 Storage Lens](#)
- [Arbeiten mit Metriken von S3 Storage Lens in CloudWatch](#)

Metriken und Dimensionen von S3 Storage Lens

Um S3-Storage-Lens-Metriken an CloudWatch zu senden, müssen Sie die CloudWatch-Veröffentlichungsoption in den fortschrittlichen Metriken und Empfehlungen von S3 Storage Lens aktivieren. Sobald erweiterte Metriken aktiviert sind, können Sie [CloudWatch-Dashboards](#) verwenden, um S3-Storage-Lens-Metriken zusammen mit anderen Anwendungsmetriken zu überwachen und eine einheitliche Ansicht auf Ihren Betriebszustand zu erstellen. Sie können Dimensionen verwenden, um Ihre S3-Storage-Lens-Metriken in CloudWatch nach Organisation, Konto, Bucket, Speicherklasse, Region und Kennzahlen-Konfigurations-ID zu filtern.

Die Metriken von S3 Storage Lens werden in CloudWatch in dem Konto veröffentlicht, das die S3-Storage-Lens-Konfiguration besitzt. Nachdem Sie die CloudWatch-Veröffentlichungsoption für erweiterte Metriken und Empfehlungen aktiviert haben, können Sie in CloudWatch auf Metriken auf Organisations-, Konto- und Bucket-Ebene zugreifen. Metriken auf Präfixebene sind in CloudWatch nicht verfügbar.

Note

S3-Storage-Lens-Metriken sind tägliche Metriken und werden einmal täglich in CloudWatch veröffentlicht. Wenn Sie S3-Storage-Lens-Metriken in CloudWatch abfragen, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3-Storage-Lens-Metriken in Ihrem S3-Storage-Lens-Dashboard in der Amazon-S3-Konsole angezeigt werden, kann es ein paar Stunden dauern, bis diese Metriken in CloudWatch angezeigt werden. Wenn Sie die CloudWatch-Veröffentlichungsoption für S3-Storage-Lens-

Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken auf CloudWatch veröffentlicht werden.

Weitere Informationen zu S3-Storage-Lens-Metriken und -Dimensionen in CloudWatch finden Sie in den folgenden Themen.

Themen

- [Metriken](#)
- [Dimensionen](#)

Metriken

In CloudWatch Metriken von S3 Storage Lens als Metriken zur Verfügung. S3-Storage-Lens-Metriken werden auf dem `AWS/S3/Storage-Lens-namespace` veröffentlicht. Dieser Namespace ist nur für S3-Storage-Lens-Metriken vorgesehen. Amazon-S3-Bucket, Anforderungs- und Replikations-Metriken werden im `AWS/S3-namespace` veröffentlicht.

Die Metriken von S3 Storage Lens werden in CloudWatch in dem Konto veröffentlicht, das die S3-Storage-Lens-Konfiguration besitzt. Nachdem Sie die CloudWatch-Veröffentlichungsoption für erweiterte Metriken und Empfehlungen aktiviert haben, können Sie in CloudWatch auf Metriken auf Organisations-, Konto- und Bucket-Ebene zugreifen. Metriken auf Präfixebene sind in CloudWatch nicht verfügbar.

In S3 Storage Lens werden Metriken aggregiert und nur in der dafür vorgesehenen Heimatregion gespeichert. Die Metriken von S3 Storage Lens werden auch in CloudWatch in der Heimatregion veröffentlicht, die Sie in der S3-Storage-Lens-Konfiguration angeben.

Eine vollständige Liste der S3-Storage-Lens-Metriken, einschließlich einer Liste der in CloudWatch verfügbaren Metriken, finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Note

Die gültige Statistik für S3-Storage-Lens-Metriken in CloudWatch ist Durchschnittlich. Weitere Informationen zu Statistiken in CloudWatch finden Sie unter [CloudWatch-Statistikdefinitionen](#) im Amazon-CloudWatch-Benutzerhandbuch.

Die Granularität von S3-Storage-Lens-Metriken in CloudWatch

S3 Storage Lens bietet Metriken für Organisation, Konto, Bucket und Präfixgranularität. S3 Storage Lens veröffentlicht S3-Storage-Lens-Metriken auf Organisations-, Konto- und Bucketebene auf CloudWatch. In CloudWatch sind S3-Storage-Lens-Metriken auf Präfixebene nicht verfügbar.

Weitere Informationen zur Granularität der in CloudWatch verfügbaren S3-Storage-Lens-Metriken finden Sie in der folgenden Liste:

- Organisation – In den Mitgliedskonten Ihrer Organisation aggregierte Metriken. S3 Storage Lens veröffentlicht Metriken für Mitgliedskonten in CloudWatch im Verwaltungskonto.
 - Organisation und Konto – Metriken für die Mitgliedskonten Ihrer Organisation.
 - Organisation und Bucket – Metriken für Amazon-S3-Buckets in den Mitgliedskonten Ihrer Organisation.
- Konto (Nicht-Organisationsebene) – Metriken, die in den Buckets Ihres Kontos zusammengefasst sind.
- Bucket (Nicht-Organisationsebene) – Metriken für einen bestimmten Bucket. In CloudWatch veröffentlicht S3 Storage Lens diese Metriken in dem AWS-Konto, das die S3-Storage-Lens-Konfiguration erstellt hat. S3 Storage Lens veröffentlicht diese Metriken nur für nicht organisatorische Konfigurationen.

Dimensionen

Wenn S3 Storage Lens Daten an CloudWatch sendet, werden jeder Metrik Dimensionen angefügt. Dimensionen sind Kategorien, die die Eigenschaften von Metriken beschreiben. Sie können Dimensionen verwenden, um die von CloudWatch zurückgegebenen Ergebnisse zu filtern.

Beispielsweise haben alle S3-Storage-Lens-Metriken in CloudWatch die `configuration_id`-Dimension. Sie können diese Dimension verwenden, um zwischen Metriken zu unterscheiden, die mit einer bestimmten S3-Storage-Lens-Konfiguration verknüpft sind. Die `organization_id` identifiziert Metriken auf Organisationsebene. Weitere Informationen zu Dimensionen der CloudWatch finden Sie unter [Dimensionen](#) im CloudWatch-Entwicklerhandbuch.

Abhängig von der Granularität der Metriken sind für S3-Storage-Lens-Metriken unterschiedliche Dimensionen verfügbar. Sie können beispielsweise die `organization_id`-Dimension zum Filtern von Metriken auf Organisationsebene nach der AWS Organizations-ID verwenden. Sie können diese Dimension jedoch nicht für Metriken auf Bucket- und Kontoebene verwenden. Weitere Informationen finden Sie unter [Metriken mit Dimensionen filtern](#).

Auf der folgenden Tabelle können Sie erkennen, welche Dimensionen für Ihre S3-Storage-Lens-Konfiguration verfügbar sind.

Dimension	Beschreibung	Bucket	Konto	Organisat	Organisat	Organisat	Organisat	Organisat
configuration_id	Der in den Metriken gemeldete Dashboard-Name für die S3-Storage-Lens-Konfiguration
metrics_version	Die Version der S3-Storage-Lens-Metriken. Die Metrikversion hat einen festen Wert von 1.0.
organization_id	Die AWS Organizations-ID der Metriken
aws_account_number	Das AWS-Konto, das den Metriken zugeordnet ist
aws_region	Die AWS-Region für die Metriken
bucket_name	Der Name des in den Metriken gemeldeten S3-Buckets
storage_class	Die Speicherklasse für den Bucket, die in den Metriken gemeldet wird
record_type	Die Granularität der Metriken: ORGANIZATION, ACCOUNT, BUCKET

Aktivieren der CloudWatch-Veröffentlichung für S3 Storage Lens

Sie können S3-Storage-Lens-Metriken in Amazon CloudWatch veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch-[Dashboards](#) zu erstellen. Sie können

auch CloudWatch-Funktionen wie Alarme und ausgelöste Aktionen, Metrikmathematik und Anomalieerkennung verwenden, um S3-Storage-Lens-Metriken zu überwachen und Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch-API-Operationen Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3-Storage-Lens-Metriken. Weitere Informationen zu CloudWatch finden Sie im [Amazon-CloudWatch-Benutzerhandbuch](#).

Die Metriken von S3 Storage Lens werden in CloudWatch in dem Konto veröffentlicht, das die S3-Storage-Lens-Konfiguration besitzt. Nachdem Sie die CloudWatch-Veröffentlichungsoption für erweiterte Metriken und Empfehlungen aktiviert haben, können Sie in CloudWatch auf Metriken auf Organisations-, Konto- und Bucket-Ebene zugreifen. Metriken auf Präfixebene sind in CloudWatch nicht verfügbar.

Sie können CloudWatch-Unterstützung für neue oder vorhandene Dashboard-Konfigurationen mithilfe der S3-Konsole, Amazon-S3-REST-APIs, AWS CLI und AWS SDKs aktivieren. Die CloudWatch-Veröffentlichungsoption ist für Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen von S3 Storage Lens aktualisiert wurden. Informationen zu Preisen für erweiterte Metriken und Empfehlungen von S3 Storage Lens finden Sie unter [Amazon-S3-Preise](#). Es fallen keine zusätzlichen Gebühren für die Veröffentlichung von CloudWatch-Metriken an. Es gelten jedoch andere CloudWatch-Gebühren z. B. für Dashboards, Alarme und API-Aufrufe.

Weitere Informationen zur Aktivierung der CloudWatch-Veröffentlichungsoption für S3-Storage-Lens-Metriken finden Sie in den folgenden Themen.

Note

S3-Storage-Lens-Metriken sind tägliche Metriken und werden einmal täglich in CloudWatch veröffentlicht. Wenn Sie S3-Storage-Lens-Metriken in CloudWatch abfragen, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3-Storage-Lens-Metriken in Ihrem S3-Storage-Lens-Dashboard in der Amazon-S3-Konsole angezeigt werden, kann es ein paar Stunden dauern, bis diese Metriken in CloudWatch angezeigt werden. Wenn Sie die CloudWatch-Veröffentlichungsoption für S3-Storage-Lens-Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken auf CloudWatch veröffentlicht werden.

Derzeit können S3-Storage-Lens-Metriken nicht über CloudWatch-Streams verwendet werden.

Verwenden der S3-Konsole

Wenn Sie ein S3-Storage-Lens-Dashboard aktualisieren, können Sie den Dashboard-Namen oder die Heimatregion nicht ändern. Sie können auch den Umfang des Standard-Dashboards nicht ändern, das auf den gesamten Speicher Ihres Kontos abgestimmt ist.

So aktualisieren Sie das S3-Storage-Lens-Dashboard, um die CloudWatch-Veröffentlichung zu aktivieren

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich S3 Storage Lens und dann Dashboards aus.
3. Wählen Sie das Dashboard aus, das Sie bearbeiten möchten, und wählen Sie dann Bearbeiten.
4. Unter Metrikauswahl, wählen Sie Fortschrittliche Metriken und Empfehlungen aus.

Fortschrittliche Metriken und Empfehlungen sind gegen Aufpreis erhältlich. Zu den erweiterten Metriken und Empfehlungen gehören ein Zeitraum von 15 Monaten für Datenabfragen, auf Präfixebene aggregierte Nutzungsmetriken, nach Bucket aggregierte Aktivitätsmetriken, die CloudWatch-Veröffentlichungsoption und kontextbezogene Empfehlungen, die Ihnen helfen, Speicherkosten zu optimieren und bewährte Methoden für den Datenschutz anzuwenden. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

5. Wählen Sie unter Fortschrittliche Metriken und Empfehlungsfunktionen auswählen die Option CloudWatch-Veröffentlichung aus.

Important

Wenn Ihre Konfiguration die Präfix-Aggregation für Nutzungsmetriken aktiviert, werden Metriken auf Präfixebene nicht in CloudWatch veröffentlicht. Auf CloudWatch werden nur S3-Storage-Lens-Metriken auf Bucket-, Konto- und Organisationsebene veröffentlicht.

6. Wählen Sie Save Changes.

So erstellen Sie ein neues S3-Storage-Lens-Dashboard, das CloudWatch-Unterstützung ermöglicht

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.


3. Klicken Sie auf Dashboard erstellen.
4. Definieren Sie unter General (Allgemein) die folgenden Konfigurationsoptionen:
 - a. Geben Sie im Feld Dashboard name (Dashboard-Name) Ihren Dashboard-Namen ein.

Dashboard-Namen müssen weniger als 65 Zeichen lang sein und dürfen keine Sonderzeichen oder Leerzeichen enthalten. Sie können den Dashboard-Namen nicht mehr ändern, nachdem Sie Ihr Dashboard erstellt haben.

- b. Wählen Sie die Heimatregion für Ihr Dashboard aus.


Metriken für alle Regionen, die in diesem Dashboard-Umfang enthalten sind, werden zentral in der angegebenen Heimatregion gespeichert. In CloudWatch sind Metriken für S3 Storage Lens auch in der Heimatregion verfügbar. Sie können die Heimatregion nicht ändern, nachdem Sie Ihr Dashboard erstellt haben.

5. (Optional) Sie fügen ein Tag hinzu, indem Sie Add tag (Tag hinzufügen) auswählen und den Schlüssel und den Wert für den Tag eingeben.

 Note

Sie können Ihrer Dashboard-Konfiguration bis zu 50 Markierungen hinzufügen.


6. Definieren Sie den Umfang für Ihre Konfiguration:
 - a. Wenn Sie eine Konfiguration auf Organisationsebene erstellen, wählen Sie die Konten aus, die in die Konfiguration aufgenommen werden sollen: Include all accounts in your configuration (Alle Konten in die Konfiguration einschließen) oder Limit the scope to your signed-in account (Umfang auf das angemeldete Konto beschränken).

 Note

Wenn Sie eine Konfiguration auf Organisationsebene erstellen, die alle Konten enthält, können Sie nur Regionen und keine Buckets ein- oder ausschließen.

- b. Wählen Sie die Regionen und Buckets aus, die S3 Storage Lens im Dashboard einschließen soll, indem Sie die folgenden Schritte ausführen:
 - Um alle Regionen einzuschließen, wählen Sie Regionen und Buckets einschließen aus.

- Um bestimmte Regionen einzubeziehen, löschen Sie Alle Regionen einschließen. Unter Wählen Sie die einzuschließenden Regionen wählen Sie die Regionen aus, die S3 Storage Lens im Dashboard enthalten soll.
- Um bestimmte Buckets einzubeziehen, löschen Sie Alle Buckets einschließen. Wählen Sie unter Einzuschließende Buckets auswählen die Buckets aus, die S3 Storage Lens in das Dashboard aufnehmen soll.

 Note


Sie können bis zu 50 Buckets auswählen.

7. Unter Metrics selection (Metrikauswahl) wählen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) aus.

Weitere Informationen zu fortschrittlichen Metriken und Preisempfehlungen finden Sie unter [Amazon-S3-Preise](#).


8. Wählen Sie unter Advanced metrics and recommendations features (Erweiterte Metrik- und Empfehlungsfunktionen) die Optionen aus, Sie aktivieren möchten:

- Advanced metrics (Erweiterte Metriken)
- CloudWatch-Veröffentlichung

 Important

Wenn Sie die Präfix-Aggregation für Ihre S3-Storage-Lens-Konfiguration aktivieren, werden Metriken auf Präfixebene nicht in CloudWatch veröffentlicht. Auf CloudWatch werden nur S3-Storage-Lens-Metriken auf Bucket-, Konto- und Organisationsebene veröffentlicht.

- Präfixzusammenfassung

 Note

Weitere Informationen über erweiterte Metriken und Empfehlungen finden Sie unter [Metrikauswahl](#).

9. Wenn Sie Advanced Metrics (Erweiterte Metriken) aktiviert haben, wählen Sie die Advanced metrics categories (Erweiterte Metrikkategorien) aus, die Sie in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten:

- Metriken für Aktivitäten
- Detailed status code metrics (Detaillierte Statuscode-Metriken)
- Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken)
- Advanced data protection metrics (Erweiterte Datensicherheitsmetriken)

Weitere Informationen zu den Metrikkategorien finden Sie unter [Metrikkategorien](#). Eine vollständige Liste der Metriken finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

10. (Optional) Konfigurieren Sie den Metrikexport.

Weitere Informationen zum Konfigurieren eines Metrikexports finden Sie unter Schritt [Erstellen eines Amazon S3-Storage-Lens-Dashboards](#).

11. Klicken Sie auf Dashboard erstellen.

Verwendung von AWS CLI

Das folgende AWS CLI-Beispiel aktiviert die CloudWatch-Veröffentlichungsoption mithilfe einer Konfiguration mit erweiterten Metriken und Empfehlungen auf Organisationsebene von S3 Storage Lens. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control put-storage-lens-configuration --account-id=555555555555 --config-id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file:///./config.json

config.json
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3 Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    }
  },
}
```

```

"AdvancedCostOptimizationMetrics": {
  "IsEnabled":true
},
"AdvancedDataProtectionMetrics": {
  "IsEnabled":true
},
"DetailedStatusCodesMetrics": {
  "IsEnabled":true
},
"BucketLevel": {
  "ActivityMetrics": {
    "IsEnabled":true //Mark this as false if you want only free metrics.
  },
  "ActivityMetrics": {
    "IsEnabled":true //Mark this as false if you want only free metrics.
  },
  "AdvancedCostOptimizationMetrics": {
    "IsEnabled":true //Mark this as false if you want only free metrics.
  },
  "DetailedStatusCodesMetrics": {
    "IsEnabled":true //Mark this as false if you want only free metrics.
  },
  "PrefixLevel":{
    "StorageMetrics":{
      "IsEnabled":true, //Mark this as false if you want only free metrics.
      "SelectionCriteria":{
        "MaxDepth":5,
        "MinStorageBytesPercentage":1.25,
        "Delimiter":"/"
      }
    }
  }
},
"Exclude": { //Replace with "Include" if you prefer to include Regions.
  "Regions": [
    "eu-west-1"
  ],
  "Buckets": [ //This attribute is not supported for AWS Organizations-level
configurations.
    "arn:aws:s3:::source_bucket1"
  ]
},
"IsEnabled": true, //Whether the configuration is enabled

```

```
"DataExport": { //Details about the metrics export
  "S3BucketDestination": {
    "OutputSchemaVersion": "V_1",
    "Format": "CSV", //You can add "Parquet" if you prefer.
    "AccountId": "111122223333",
    "Arn": "arn:aws:s3:::destination-bucket-name", // The destination bucket for your
metrics export must be in the same Region as your S3 Storage Lens configuration.
    "Prefix": "prefix-for-your-export-destination",
    "Encryption": {
      "SSES3": {}
    }
  },
  "CloudWatchMetrics": {
    "IsEnabled": true //Mark this as false if you want to export only free metrics.
  }
}
```

Verwenden des AWS-SDK für Java

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
```

```
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination
        bucket for your metrics export must be in the same Region as your S3 Storage Lens
        configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
        abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
            PrefixLevelStorageMetrics()
                .withIsEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                .withAdvancedCostOptimizationMetrics(new
            AdvancedCostOptimizationMetrics().withIsEnabled(true))
                .withAdvancedDataProtectionMetrics(new
            AdvancedDataProtectionMetrics().withIsEnabled(true))
                .withDetailedStatusCodesMetrics(new
            DetailedStatusCodesMetrics().withIsEnabled(true))
                .withPrefixLevel(new
            PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                .withAdvancedCostOptimizationMetrics(new
            AdvancedCostOptimizationMetrics().withIsEnabled(true))
```

```
        .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
        .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
        .withBucketLevel(bucketLevel);

Include include = new Include()
    .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
    .withRegions(Arrays.asList("us-west-2"));

StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
    .withSSES3(new SSES3());
S3BucketDestination s3BucketDestination = new S3BucketDestination()
    .withAccountId(exportAccountId)
    .withArn(exportBucketArn)
    .withEncryption(exportEncryption)
    .withFormat(exportFormat)
    .withOutputSchemaVersion(OutputSchemaVersion.V_1)
    .withPrefix("Prefix");
CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
    .withIsEnabled(true);
StorageLensDataExport dataExport = new StorageLensDataExport()
    .withCloudWatchMetrics(cloudWatchMetrics)
    .withS3BucketDestination(s3BucketDestination);

StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
    .withArn(awsOrgARN);

StorageLensConfiguration configuration = new StorageLensConfiguration()
    .withId(configurationId)
    .withAccountLevel(accountLevel)
    .withInclude(include)
    .withDataExport(dataExport)
    .withAwsOrg(awsOrg)
    .withIsEnabled(true);

List<StorageLensTag> tags = Arrays.asList(
    new StorageLensTag().withKey("key-1").withValue("value-1"),
    new StorageLensTag().withKey("key-2").withValue("value-2")
);

AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
    .withCredentials(new ProfileCredentialsProvider())
```

```
        .withRegion(US_WEST_2)
        .build();

        s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
            .withAccountId(sourceAccountId)
            .withConfigId(configurationId)
            .withStorageLensConfiguration(configuration)
            .withTags(tags)
        );
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

Verwenden der REST-API

Wenn Sie die CloudWatch-Veröffentlichungsoption mit der Amazon-S3-REST-API aktivieren möchten, können Sie [PutStorageLensConfiguration](#) verwenden.

Nächste Schritte

Nachdem Sie die CloudWatch-Veröffentlichungsoption aktiviert haben, können Sie in CloudWatch auf Ihre S3-Storage-Lens-Metriken zugreifen. Sie können CloudWatch-Funktionen auch nutzen, um Ihre S3-Storage-Lens-Daten in CloudWatch zu überwachen und zu analysieren. Weitere Informationen finden Sie unter den folgenden Themen:

- [Metriken und Dimensionen von S3 Storage Lens](#)
- [Arbeiten mit Metriken von S3 Storage Lens in CloudWatch](#)

Arbeiten mit Metriken von S3 Storage Lens in CloudWatch

Sie können S3-Storage-Lens-Metriken in Amazon CloudWatch veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch-[Dashboards](#) zu erstellen. Sie können auch CloudWatch-Funktionen wie Alarme und ausgelöste Aktionen, Metrikmathematik und

Anomalieerkennung verwenden, um S3-Storage-Lens-Metriken zu überwachen und Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch-API-Operationen Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3-Storage-Lens-Metriken. Weitere Informationen zu CloudWatch finden Sie im [Amazon-CloudWatch-Benutzerhandbuch](#).

Sie können die CloudWatch-Veröffentlichungsoption für neue oder vorhandene Dashboard-Konfigurationen mithilfe der Amazon-S3-Konsole, der Amazon-S3-REST-APIs, AWS CLI und AWS SDKs aktivieren. Die CloudWatch-Veröffentlichungsoption ist für Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen von S3 Storage Lens aktualisiert wurden. Informationen zu Preisen für erweiterte Metriken und Empfehlungen von S3 Storage Lens finden Sie unter [Amazon-S3-Preise](#). Es fallen keine zusätzlichen Gebühren für die Veröffentlichung von CloudWatch-Metriken an. Es gelten jedoch andere CloudWatch-Gebühren z. B. für Dashboards, Alarme und API-Aufrufe. Weitere Informationen hierzu finden Sie unter [Amazon CloudWatch – Preise](#).

Die Metriken von S3 Storage Lens werden in CloudWatch in dem Konto veröffentlicht, das die S3-Storage-Lens-Konfiguration besitzt. Nachdem Sie die CloudWatch-Veröffentlichungsoption für erweiterte Metriken und Empfehlungen aktiviert haben, können Sie in CloudWatch auf Metriken auf Organisations-, Konto- und Bucket-Ebene zugreifen. Metriken auf Präfixebene sind in CloudWatch nicht verfügbar.

Note

S3-Storage-Lens-Metriken sind tägliche Metriken und werden einmal täglich in CloudWatch veröffentlicht. Wenn Sie S3-Storage-Lens-Metriken in CloudWatch abfragen, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3-Storage-Lens-Metriken in Ihrem S3-Storage-Lens-Dashboard in der Amazon-S3-Konsole angezeigt werden, kann es ein paar Stunden dauern, bis diese Metriken in CloudWatch angezeigt werden. Wenn Sie die CloudWatch-Veröffentlichungsoption für S3-Storage-Lens-Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken auf CloudWatch veröffentlicht werden.

Derzeit können S3-Storage-Lens-Metriken nicht über CloudWatch-Streams verwendet werden.

Weitere Informationen zum Arbeiten mit S3-Storage-Lens-Metriken in CloudWatch finden Sie in den folgenden Themen.

Themen

- [Arbeiten mit CloudWatch-Dashboards](#)
- [Alarmer einstellen, Aktionen auslösen und Anomalieerkennung verwenden](#)
- [Metriken mit Dimensionen filtern](#)
- [Berechnen neuer Metriken mit Metrikmathematik](#)
- [Verwenden von Suchausdrücken in Diagrammen](#)

Arbeiten mit CloudWatch-Dashboards

Sie können Cloudwatch-Dashboards verwenden, um S3-Storage-Lens-Metriken zusammen mit anderen Anwendungsmetriken zu überwachen und eine einheitliche Sicht auf Ihren Betriebszustand zu erstellen. Dashboards sind eine anpassbare Homepage in der CloudWatch-Konsole, die Sie verwenden können, um Ihre -Ressourcen in einer zentralen Ansicht zu überwachen.

CloudWatch verfügt über eine breite Berechtigungssteuerung, die die Beschränkung des Zugriffs auf bestimmte Metriken oder Dimensionen nicht unterstützt. Benutzer in Ihrem Konto oder Ihrer Organisation, die Zugriff auf CloudWatch haben, haben Zugriff auf Metriken für alle S3-Storage-Lens-Konfigurationen, bei denen die CloudWatch-Supportoption aktiviert ist. Sie können Berechtigungen für bestimmte Dashboards nicht wie in S3 Storage Lens verwalten. Weitere Informationen zu CloudWatch-Berechtigungen finden Sie unter [Verwalten von Zugriffsberechtigungen für Ihre CloudWatch-Ressourcen](#) im Amazon-CloudWatch-Benutzerhandbuch.

Weitere Informationen zur Verwendung von CloudWatch-Dashboards und zum Konfigurieren von Berechtigungen finden Sie unter [Verwenden von Amazon-CloudWatch-Dashboards](#) und [Freigeben von CloudWatch-Dashboards](#) im Amazon-CloudWatch-Benutzerhandbuch.

Alarmer einstellen, Aktionen auslösen und Anomalieerkennung verwenden

Sie können CloudWatch-Alarmer konfigurieren, die S3-Storage-Lens-Metriken in CloudWatch ansehen, und Maßnahmen ergreifen, wenn ein Schwellenwert überschritten wird. Sie können beispielsweise einen Alarm konfigurieren, der eine Amazon-SNS-Benachrichtigung sendet, wenn die Metrik Incomplete Multipart Upload Bytes (Unvollständige Bytes für mehrteilige Uploads) an drei aufeinanderfolgenden Tagen 1 GB überschreitet.

Sie können auch die Anomalieerkennung aktivieren, um Ihre S3-Storage-Lens-Metriken kontinuierlich zu analysieren und normale Baselines und Oberflächenanomalien zu ermitteln. Sie können Anomalieerkennungsalarmer basierend auf dem erwarteten Wert einer Metrik erstellen. Sie können beispielsweise Anomalien für die Metrik Object Lock Enabled Bytes (Bytes mit aktivierter Objektsperre), um das unbefugte Entfernen von Objektsperreinstellungen zu erkennen.

Weitere Informationen und Beispiele finden Sie unter [Verwenden von Amazon-CloudWatch-Alarmen](#) und [Erstellen eines Alarms aus einer Metrik in einem Diagramm](#) im Amazon-CloudWatch-Benutzerhandbuch.

Metriken mit Dimensionen filtern

Sie können Dimensionen verwenden, um S3-Storage-Lens-Metriken in der CloudWatch-Konsole zu filtern. Beispielsweise können Sie nach `configuration_id`, `aws_account_number`, `aws_region`, `bucket_name` und mehr filtern.

S3 Storage Lens unterstützt mehrere Dashboard-Konfigurationen pro Konto. Dies bedeutet, dass verschiedene Konfigurationen denselben Bucket enthalten können. Wenn diese Metriken in CloudWatch veröffentlicht werden, enthält der Bucket doppelte Metriken in CloudWatch. Um Metriken nur für eine bestimmte S3-Storage-Lens-Konfiguration in CloudWatch anzuzeigen, können Sie die `configuration_id`-Dimension verwenden. Wenn Sie nach `configuration_id` filtern, sehen Sie nur Metriken, die mit der Konfiguration verknüpft sind, die Sie identifizieren.

Weitere Informationen zum Filtern nach Konfigurations-ID finden Sie unter [Suche nach verfügbaren Metriken](#) im Amazon-CloudWatch-Benutzerhandbuch.

Berechnen neuer Metriken mit Metrikmathematik

Mithilfe von Metrikberechnungen können Sie mehrere Metriken mit S3 Storage Lens abfragen und mathematische Ausdrücke verwenden, um neue Zeitreihen basierend auf diesen Metriken zu erstellen. Sie können beispielsweise eine neue Metrik für unverschlüsselte Objekte erstellen, indem Sie verschlüsselte Objekte von der Objektanzahl subtrahieren. Sie können auch eine Metrik erstellen, um die durchschnittliche Objektgröße zu erhalten, indem Sie `StorageBytes` durch `ObjectCount` oder die Anzahl der Bytes dividieren, auf die an einem Tag zugegriffen wird, indem Sie `BytesDownloaded` durch `StorageBytes` dividieren.

Weitere Informationen finden Sie unter [Metrikmathematik verwenden](#) im Amazon-CloudWatch-Benutzerhandbuch.

Verwenden von Suchausdrücken in Diagrammen

Mit S3-Storage-Lens-Metriken können Sie einen Suchausdruck erstellen. Sie können beispielsweise einen Suchausdruck für alle Metriken namens `IncompleteMultipartUploadStorageBytes` erstellen und dem Ausdruck `SUM` hinzufügen. Mit diesem Suchausdruck können Sie Ihre gesamten unvollständigen mehrteiligen Upload-Bytes über alle Dimensionen Ihres Speichers in einer einzigen Metrik sehen.

Dieses Beispiel zeigt die Syntax, die Sie verwenden würden, um einen Suchausdruck für alle Metriken mit dem Namen `IncompleteMultipartUploadStorageBytes` zu erstellen.

```
SUM(SEARCH( '{AWS/S3/Storage-Lens,aws_account_number,aws_region,configuration_id,metrics_version,record_type,storage_class}'  
MetricName="IncompleteMultipartUploadStorageBytes"', 'Average',86400))
```

Weitere Informationen zu dieser Syntax finden Sie unter [Syntax für CloudWatch-Suchausdrücke](#) im Amazon-CloudWatch-Benutzerhandbuch. Informationen zum Erstellen eines CloudWatch-Diagramms mit einem Suchausdruck finden Sie unter [Erstellen eines CloudWatch-Diagramms mit einem Suchausdruck](#) im Amazon-CloudWatch-Benutzerhandbuch.

Anwendungsfälle für Metriken von Amazon S3 Storage Lens

Sie können Ihr Dashboard von Amazon S3 Storage Lens verwenden, um Erkenntnisse und Trends zu visualisieren, Ausreißer zu markieren und Empfehlungen einzusehen. Die Metriken von S3 Storage Lens sind in Kategorien unterteilt, die den wichtigsten Anwendungsfällen entsprechen. Diese Metriken bieten Ihnen folgende Möglichkeiten:

- Identifizieren von Möglichkeiten zur Kostenoptimierung
- Anwenden von bewährten Datenschutzmethoden
- Anwenden von bewährten Zugriffsverwaltungsmethoden
- Verbessern der Leistung von Anwendungs-Workloads

Mithilfe der Metriken zur Kostenoptimierung können Sie Möglichkeiten ermitteln, wie Sie Ihre Amazon-S3-Speicherkosten senken können. Sie können Buckets mit mehrteiligen Uploads identifizieren, die älter als 7 Tage sind, oder Buckets, in denen sich nicht aktuelle Versionen ansammeln.

In ähnlicher Weise können Sie Datenschutzmetriken verwenden, um Buckets zu identifizieren, die nicht den bewährten Datenschutzmethoden Ihres Unternehmens entsprechen. Sie können beispielsweise Buckets identifizieren, die keine AWS Key Management Service-Schlüssel (SSE-KMS) für die Standardverschlüsselung verwenden oder für die S3 Versioning nicht aktiviert ist.

Mit den Zugriffsverwaltungsmetriken von S3 Storage Lens können Sie Bucket-Einstellungen für S3 Object Ownership identifizieren, sodass Sie Zugriffssteuerungslisten (ACLs) Berechtigungen zu Bucket-Richtlinien migrieren und ACLs deaktivieren können.

Wenn Sie die Option [S3 Storage Lens advanced metrics](#) (Erweiterte Metriken von S3 Storage Lens) aktiviert haben, können Sie die detaillierten Statuscode-Metriken verwenden, um die Anzahl von erfolgreichen oder fehlgeschlagenen Anforderungen zu erhalten. Anhand dieser Informationen können Sie Zugriffs- oder Leistungsprobleme beheben.

Mithilfe erweiterter Metriken können Sie auch auf zusätzliche Metriken zur Kostenoptimierung und zum Datenschutz zugreifen, anhand derer Sie Möglichkeiten zur weiteren Senkung Ihrer Gesamtkosten für den S3-Speicher identifizieren und besser an bewährte Methoden für den Schutz Ihrer Daten anpassen können. Erweiterte Metriken zur Kostenoptimierung umfassen beispielsweise die Anzahl von Lebenszyklusregeln, mit der Sie Buckets identifizieren können, für die keine Lebenszyklusregeln festgelegt sind, um unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen zu lassen. Zu den erweiterten Datenschutzmetriken gehört die Anzahl der Replikationsregeln.

Weitere Informationen zu den Metrikkategorien finden Sie unter [Metrikkategorien](#). Eine vollständige Liste der Metriken von S3 Storage Lens finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Themen

- [Verwenden von Amazon S3 Storage Lens zur Optimierung Ihrer Speicherkosten](#)
- [Verwenden von S3 Storage Lens zum Schutz Ihrer Daten](#)
- [Verwenden von S3 Storage Lens zur Überwachung der Einstellungen für die Objekteigentümerschaft](#)
- [Verwenden der Metriken von S3 Storage Lens zur Verbesserung der Leistung](#)

Verwenden von Amazon S3 Storage Lens zur Optimierung Ihrer Speicherkosten

Sie können die Kostenoptimierungsmetriken von S3 Storage Lens verwenden, um die Gesamtkosten Ihres S3-Speichers zu senken. Mithilfe von Metriken zur Kostenoptimierung können Sie überprüfen, dass Sie Amazon S3 kostengünstig und gemäß den bewährten Methoden konfiguriert haben. Sie können beispielsweise die folgenden Möglichkeiten zur Kostenoptimierung identifizieren:

- Buckets mit unvollständigen mehrteiligen Uploads, die älter als 7 Tage sind
- Buckets, in denen sich zahlreiche nicht aktuelle Versionen ansammeln
- Buckets, die keine Lebenszyklusregeln zum Abbruch unvollständiger mehrteiliger Uploads haben
- Buckets, die keine Lebenszyklusregeln für das Ablaufen von Objekten mit nicht aktuellen Versionen haben

- Buckets, die keine Lebenszyklusregeln für den Übergang von Objekten in eine andere Speicherklasse haben

Sie können diese Daten dann verwenden, um Ihren Buckets zusätzliche Lebenszyklusregeln hinzuzufügen.

In den folgenden Beispielen wird gezeigt, wie Sie Metriken zur Kostenoptimierung in Ihrem S3-Storage-Lens-Dashboard verwenden, um Ihre Speicherkosten zu optimieren.

Themen

- [Identifizieren Sie Ihre größten S3-Buckets](#)
- [Entdecken Sie kalte Amazon S3-Buckets](#)
- [Suchen von unvollständigen mehrteiligen Uploads](#)
- [Reduzieren Sie die Anzahl der beibehaltenen nicht aktuellen Versionen](#)
- [Identifizieren von Buckets ohne Lebenszyklusregeln und Überprüfen der Anzahl von Lebenszyklusregeln](#)

Identifizieren Sie Ihre größten S3-Buckets

Sie zahlen für das Speichern von Objekten in S3-Buckets. Der Ihnen in Rechnung gestellte Preis hängt von der Größe Ihrer Objekte, der Speicherdauer der Objekte und ihren Speicherklassen ab. Mit S3 Storage Lens erhalten Sie einen zentralen Überblick über alle Buckets in Ihrem Konto. Um alle Buckets in allen Konten Ihrer Organisation anzuzeigen, können Sie ein AWS Organizations-Grad S3-Storage-Lens-Dashboard konfigurieren. In dieser Dashboard-Ansicht können Sie Ihre größten Buckets identifizieren.


Schritt 1: Identifizieren Ihrer größten Buckets

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.

Wenn das Dashboard geöffnet wird, können Sie das Datum sehen, für das S3 Storage Lens zuletzt Metriken gesammelt hat. Ihr Dashboard wird immer für das letzte Datum geladen, für das Metriken verfügbar sind.

4. Scrollen Sie zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum), um eine Rangfolge Ihrer größten Buckets nach der Metrik Total storage (Gesamtpeicher) für einen ausgewählten Zeitraum anzuzeigen.

Sie können die Sortierreihenfolge ändern, um die kleinsten Buckets anzuzeigen. Sie können auch die Auswahl Metric (Metrik) anpassen, um Ihre Buckets nach einer der verfügbaren Metriken zu ordnen. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

 Note

Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter [Metrikauswahl](#).

5. Für detailliertere Informationen zu Ihren Buckets scrollen Sie zum Seitenanfang und wählen Sie dann die Registerkarte Bucket.

Auf der Registerkarte Buckets können Sie Details wie die aktuelle Wachstumsrate, die durchschnittliche Objektgröße, die größten Präfixe und die Anzahl der Objekte anzeigen.

Schritt 2: Navigieren zu Ihren Buckets und Untersuchen von Daten

Nachdem Sie Ihre größten S3-Buckets identifiziert haben, können Sie innerhalb der S3-Konsole zu jedem Bucket navigieren, um seine Objekte anzuzeigen, die damit verbundene Workload zu verstehen und interne Eigentümer des Buckets zu identifizieren. Sie können die Bucket-Eigentümer kontaktieren, um herauszufinden, ob das Wachstum erwartet wird oder ob es einer weiteren Überwachung und Kontrolle bedarf.

Entdecken Sie kalte Amazon S3-Buckets

Wenn Sie die [fortschrittlichen Metriken von S3 Storage Lens](#) aktiviert haben, können Sie [Aktivitätsmetriken](#) verwenden, um zu verstehen, wie kalt Ihre S3-Buckets sind. Ein „kalter“ Bucket ist ein Bucket, auf dessen Speicher nicht mehr (oder sehr selten) zugegriffen wird. Dieser Mangel an Aktivität weist normalerweise darauf hin, dass auf die Objekte des Buckets nicht häufig zugegriffen wird.

Aktivitätsmetriken wie GET-Anforderungen und Download-Bytes geben an, wie oft täglich auf Ihre Buckets zugegriffen wird. Um die Konsistenz des Zugriffsmusters zu verstehen und Buckets zu erkennen, auf die überhaupt nicht mehr zugegriffen wird, können Sie diese Daten über mehrere Monate hinweg trenden. Die Metrik der Abruftrate, die als Download-Bytes/Gesamtspeicher berechnet wird, gibt den Anteil des Speichers in einem Bucket an, auf den täglich zugegriffen wird.

Note

Download-Bytes werden dupliziert, wenn das gleiche Objekt mehrmals am Tag heruntergeladen wird.

Voraussetzung

Wenn Sie Aktivitätsmetriken in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten, müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Activity metrics (Aktivitätsmetriken) auswählen. Weitere Informationen finden Sie unter [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#).

Schritt 1: Identifizieren aktiver Buckets

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie die Registerkarte Bucket aus und scrollen Sie dann nach unten zum Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum).

Im Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) können Sie Ihre Buckets in mehreren Dimensionen darstellen, indem Sie drei beliebige Metriken verwenden, um die X-axis (x-Achse), Y-axis (y-Achse) und die Size (Größe) der Blase darzustellen.

5. Wählen Sie für X-axis (x-Achse) Y-axis (y-Achse) und Size (Größe) die Metriken Total storage (Gesamtspeicher), % retrieval rate (% Abruftrate) und Average object size (Durchschnittliche Objektgröße) aus, um „kalte“ Buckets zu finden.

6. Suchen Sie im Abschnitt *Bubble analysis by buckets for date* (Blasenanalyse nach Buckets für Datum) nach Buckets mit Abrufzeiten von null (oder nahe null) und einer größeren relativen Speichergröße. Wählen Sie dann die Blase aus, die den Bucket darstellt.

Es erscheint ein Feld mit Auswahlmöglichkeiten für detailliertere Einblicke. Führen Sie eine der folgenden Aktionen aus:

- a. Damit die Registerkarte Bucket aktualisiert wird und nur Metriken für den ausgewählten Bucket anzeigt, wählen Sie *Drill down* (Aufgliedern) und dann *Apply* (Anwenden) aus.
- b. Wenn Sie Ihre Daten auf Bucket-Ebene nach Konto, AWS-Region, Speicherklasse oder Bucket aggregieren möchten, wählen Sie *Analyze by* (Analysieren nach) aus und treffen Sie dann eine Auswahl für Dimension. Wenn Sie beispielsweise nach Speicherklasse aggregieren möchten, wählen Sie *Storage class* (Speicherklasse) für Dimension aus.

Um Buckets zu finden, die kalt geworden sind, führen Sie eine Blasenanalyse mit den Metriken Gesamtspeicher, Abrufzeit in % und Durchschnittliche Objektgröße durch. Suchen Sie nach Buckets mit Abrufzeiten von null (oder nahe null) und einer größeren relativen Speichergröße.

Die Registerkarte Bucket Ihres Dashboards wird aktualisiert und zeigt Daten für die von Ihnen ausgewählte Aggregation oder den ausgewählten Filter an. Wenn Sie nach Speicherklasse oder einer anderen Dimension aggregiert haben, wird diese neue Registerkarte in Ihrem Dashboard geöffnet (z. B. die Registerkarte *Storage class* (Speicherklasse)).

Schritt 2: Untersuchen kalter Buckets

Von hier aus können Sie die Eigentümer von kalten Buckets in Ihrem Konto oder Ihrer Organisation identifizieren und herausfinden, ob dieser Speicher noch benötigt wird. Anschließend können Sie die Kosten optimieren, indem Sie [Ablaufkonfigurationen für den Lebenszyklus](#) für diese Buckets konfigurieren oder die Daten in einer der [Amazon S3 Glacier-Speicherklassen](#) archivieren.

Um das Problem von kalten Buckets in Zukunft zu vermeiden, können Sie Ihre [Daten mithilfe von S3-Lifecycle-Konfigurationen für Ihre Buckets automatisch umstellen](#) oder die [automatische Archivierung mit S3 Intelligent-Tiering](#) aktivieren.

Sie können Schritt 1 auch verwenden, um „heiße“ Buckets zu identifizieren. Anschließend können Sie sich vergewissern, dass diese Buckets die richtige [S3-Speicherklasse](#) verwenden, um sicherzustellen, dass sie ihre Anforderungen in Bezug auf Leistung und Kosten am effektivsten bearbeiten.

Suchen von unvollständigen mehrteiligen Uploads

Sie können mehrteilige Uploads verwenden, um sehr große Objekte (bis zu 5 TB) als einen Satz von Teilen hochzuladen und so den Durchsatz zu verbessern und Netzwerkprobleme schneller zu beheben. In Fällen, in denen der mehrteilige Upload-Prozess nicht abgeschlossen wird, verbleiben die unvollständigen Teile im Bucket (in einem unbrauchbaren Zustand). Für diese unvollständigen Teile fallen Speicherkosten an, bis der Upload-Prozess abgeschlossen ist oder die unvollständigen Teile entfernt werden. Weitere Informationen finden Sie unter [Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#).

Mit S3 Storage Lens können Sie die Anzahl der unvollständigen mehrteiligen Upload-Bytes in Ihrem Konto oder in Ihrem gesamten Unternehmen ermitteln. Dies umfasst auch mehrteilige Uploads, die älter sind als 7 Tage. Eine vollständige Liste der Metriken für unvollständige mehrteilige Uploads finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Als bewährte Methode empfehlen wir, Lebenszyklusregeln so zu konfigurieren, dass unvollständige mehrteilige Uploads, die älter als eine bestimmte Anzahl von Tagen sind, ablaufen. Wenn Sie Ihre Lebenszyklusregel so erstellen, dass unvollständige mehrteilige Uploads ablaufen, empfehlen wir 7 Tage als geeigneten Ausgangspunkt.

Schritt 1: Überprüfen allgemeiner Trends für unvollständige mehrteilige Uploads

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot for date (Snapshot für Datum) unter Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus.

Der Abschnitt Snapshot for date (Snapshot für Datum) wird aktualisiert und zeigt nun Metriken zu Cost optimization (Kostenoptimierung) an. Dazu gehören Incomplete multipart upload bytes greater than 7 days old (Unvollständige mehrteilige Upload-Bytes, die älter als 7 Tage sind).

In jedem Diagramm Ihres S3-Storage-Lens-Dashboards können Sie Metriken für unvollständige mehrteilige Uploads sehen. Sie können diese Metriken verwenden, um die Auswirkungen unvollständiger mehrteiliger Upload-Bytes auf Ihren Speicher weiter zu bewerten, einschließlich deren Beitrag zu den allgemeinen Wachstumstrends. Sie können auch eine detaillierte Aufschlüsselung tieferer Aggregationsebenen aufrufen. Verwenden Sie dazu die Registerkarten

Account (Konto), AWS-Region, Bucket oder Storage class (Speicherklasse), um Ihre Daten eingehender zu analysieren. Ein Beispiel finden Sie unter [Entdecken Sie kalte Amazon S3-Buckets](#).

Schritt 2: Identifizieren von Buckets, die die unvollständigsten mehrteiligen Upload-Bytes, aber keine Lebenszyklusregeln zum Abbruch von unvollständigen mehrteiligen Uploads haben

Voraussetzung

Wenn Sie in Ihrem S3-Storage-Lens-Dashboard die Metrik Abort incomplete multipart upload lifecycle rule count (Anzahl der Lebenszyklusregeln zum Abbrechen unvollständiger mehrteiliger Uploads) sehen möchten, müssen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken) auswählen. Weitere Informationen finden Sie unter [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#).

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wenn Sie bestimmte Buckets identifizieren möchten, in denen sich unvollständige mehrteilige Uploads ansammeln, die älter als 7 Tage sind, gehen Sie zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum).

Standardmäßig werden im Abschnitt Top N overview for date (Top-N-Übersicht für Datum) Metriken für die drei wichtigsten Buckets angezeigt. Sie können die Anzahl der Buckets im Feld Top N erhöhen oder verringern. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. (Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.)

Note

Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter [Metrikauswahl](#).

5. Wählen Sie für Metric (Metrik) in der Kategorie Cost optimization (Kostenoptimierung) die Option Incomplete multipart upload bytes greater than 7 days old (Unvollständige mehrteilige Upload-Bytes, die älter als 7 Tage sind) aus.

Unter Top number buckets (Top N-Buckets) sehen Sie die Buckets mit den unvollständigsten mehrteiligen Upload-Speicherbytes, die älter als 7 Tage sind.

6. Wenn Sie detailliertere Metriken auf Bucket-Ebene für unvollständige mehrteilige Uploads anzeigen möchten, scrollen Sie zum Seitenanfang und wählen Sie dann die Registerkarte Bucket aus.
7. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Cost optimization (Kostenoptimierung) für die angezeigten Buckets.

8. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Kostenoptimierungsmetriken angezeigt werden, wählen Sie das Präferenzensymbol



aus.

9. Deaktivieren Sie die Schalter für alle Metriken zur Kostenoptimierung, bis nur noch Incomplete multipart upload bytes greater than 7 days old (Unvollständige mehrteilige Upload-Bytes, die älter als 7 Tage sind) und Abort incomplete multipart upload lifecycle rule count (Anzahl der Lebenszyklusregeln zum Abbrechen unvollständiger mehrteiliger Uploads) ausgewählt sind.
10. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
11. Wählen Sie Bestätigen aus.

Die Liste Buckets wird aktualisiert und zeigt Metriken auf Bucket-Ebene für unvollständige mehrteilige Uploads und die Anzahl der Lebenszyklusregeln an. Sie können diese Daten verwenden, um Buckets zu identifizieren, die die unvollständigsten mehrteiligen Upload-Bytes enthalten, die älter als 7 Tage sind und denen Lebenszyklusregeln fehlen, um unvollständige mehrteilige Uploads abubrechen. Anschließend können Sie in der S3-Konsole zu diesen Buckets navigieren und Lebenszyklusregeln hinzufügen, um abgebrochene, unvollständige mehrteilige Uploads zu löschen.

Schritt 3: Hinzufügen einer Lebenszyklusregel, um unvollständige mehrteilige Uploads nach 7 Tagen zu löschen

Um unvollständige mehrteilige Uploads automatisch zu verwalten, können Sie die S3-Konsole verwenden, um eine Lebenszykluskonfiguration zu erstellen und unvollständige mehrteilige Upload-Bytes aus einem Bucket nach einer bestimmten Anzahl von Tagen ablaufen zu lassen. Weitere Informationen finden Sie unter [Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads](#).

Reduzieren Sie die Anzahl der beibehaltenen nicht aktuellen Versionen

Wenn die S3 Versioning aktiviert ist, werden mehrere Versionen desselben Objekts beibehalten, die zur schnellen Wiederherstellung von Daten verwendet werden können, wenn ein Objekt versehentlich gelöscht oder überschrieben wird. Wenn Sie S3 Versioning aktiviert haben, ohne Lebenszyklusregeln für die Umstellung oder den Ablauf nicht aktueller Versionen zu konfigurieren, kann sich eine große Anzahl nicht aktueller Versionen ansammeln, was sich auf die Speicherkosten auswirken kann. Weitere Informationen finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Schritt 1: Identifizieren von Buckets mit den meisten nicht aktuellen Objektversionen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot for date (Snapshot für Datum) unter Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus.


Der Abschnitt Snapshot for date (Snapshot für Datum) wird aktualisiert und zeigt nun Metriken für Cost optimization (Kostenoptimierung) an, zu denen auch die Metrik % noncurrent version bytes (% Bytes der nicht aktuellen Version) gehört. Die Metrik % noncurrent version bytes (% Bytes der nicht aktuellen Version) stellt den Anteil Ihrer gesamten Speicherbytes dar, der im Rahmen des Dashboards und für das ausgewählte Datum auf nicht aktuelle Versionen zurückzuführen ist.

Note

Wenn der Wert % noncurrent version bytes (% Bytes der nicht aktuellen Version) mehr als 10 % Ihres Speicherplatzes auf Kontoebene beträgt, könnte dies ein Hinweis darauf sein, dass Sie zu viele Versionen speichern.

5. So identifizieren Sie bestimmte Buckets, in denen sich eine große Anzahl nicht aktueller Versionen ansammelt:
 - a. Scrollen Sie nach unten zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum). Geben Sie für Top N die Anzahl der Buckets ein, deren Daten Sie sehen möchten.
 - b. Wählen Sie für Metric (Metrik) die Option % noncurrent version bytes (% Bytes der nicht aktuellen Version) aus.

Unter Top number buckets (Top N-Buckets) werden die Buckets (für die von Ihnen angegebene Anzahl) mit dem höchsten Wert für % noncurrent version bytes (% Bytes der nicht aktuellen Version) angezeigt. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

 Note

Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter [Metrikauswahl](#).

- c. Wenn Sie detailliertere Metriken auf Bucket-Ebene für nicht aktuelle Objektversionen anzeigen möchten, scrollen Sie zum Seitenanfang und wählen Sie dann die Registerkarte Bucket aus.

In jedem Diagramm oder jeder Visualisierung Ihres S3-Storage-Lens-Dashboards können Sie mithilfe der Registerkarten Account (Konto), AWS-Region, Storage classe (Speicherklasse) oder Bucket tiefere Aggregationsebenen aufrufen. Ein Beispiel finden Sie unter [Entdecken Sie kalte Amazon S3-Buckets](#).

- d. Wählen Sie im Abschnitt Buckets für Metric categories (Metrikkategorien) die Option Cost optimization (Kostensoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

Sie können jetzt die Metrik % noncurrent version bytes (% Bytes der nicht aktuellen Version) zusammen mit anderen Metriken sehen, die sich auf nicht aktuelle Versionen beziehen.


Schritt 2: Identifizieren von Buckets, für die Lebenszyklusregeln für Übergang und Ablauf nicht aktueller Versionen fehlen

Voraussetzung

Wenn Sie in Ihrem S3-Storage-Lens-Dashboard die Metriken Noncurrent version transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang nicht aktueller Versionen) und Noncurrent version expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf nicht aktueller Versionen) zu sehen, müssen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken) auswählen. Weitere Informationen finden Sie unter [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#).

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Cost optimization (Kostenoptimierung) für die angezeigten Buckets.

6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Kostenoptimierungsmetriken angezeigt werden, wählen Sie das Präferenzensymbol  aus.)
7. Deaktivieren Sie die Schalter für alle Metriken zur Kostenoptimierung, bis nur noch die folgenden Metriken ausgewählt sind:

- % noncurrent version bytes (% Bytes der nicht aktuellen Version)
- Noncurrent version transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang nicht aktueller Versionen)
- Noncurrent version expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf nicht aktueller Versionen)

8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Bestätigen aus.

Die Liste Buckets wird aktualisiert und zeigt Metriken für die Anzahl der Bytes nicht aktueller Versionen und die Anzahl der Lebenszyklusregeln für nicht aktuelle Versionen an. Sie können diese Daten verwenden, um Buckets zu identifizieren, die einen hohen Prozentsatz an Bytes nicht aktueller Versionen enthalten, für die jedoch Übergangs- und Ablauflebenszyklusregeln fehlen. Anschließend können Sie in der S3-Konsole zu diesen Buckets navigieren und ihnen Lebenszyklusregeln hinzufügen.

Schritt 3: Hinzufügen von Lebenszyklusregeln für den Übergang oder den Ablauf nicht aktueller Objektversionen

Nachdem Sie festgestellt haben, welche Buckets eine weitere Untersuchung erfordern, können Sie in der S3-Konsole zu den Buckets navigieren und eine Lebenszyklusregel hinzufügen, um nicht aktuelle Versionen nach einer bestimmten Anzahl von Tagen ablaufen zu lassen. Um die Kosten zu senken und gleichzeitig nicht aktuelle Versionen beizubehalten, können Sie alternativ eine Lebenszyklusregel konfigurieren, um nicht aktuelle Versionen auf eine der Amazon S3 Glacier-Speicherklassen zu übertragen. Weitere Informationen finden Sie unter [Beispiel 6: Spezifikation einer Lebenszyklus-Konfigurationsregel für einen Bucket mit Versioning](#).

Identifizieren von Buckets ohne Lebenszyklusregeln und Überprüfen der Anzahl von Lebenszyklusregeln

S3 Storage Lens bietet Metriken zur Anzahl der S3-Lebenszyklusregeln, anhand derer Sie Buckets identifizieren können, denen Lebenszyklusregeln fehlen. Um Buckets zu finden, die keine Lebenszyklusregeln haben, können Sie die Metrik Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln) verwenden. Ein Bucket ohne S3-Lifecycle-Konfiguration verfügt möglicherweise über Speicher, den Sie nicht mehr benötigen oder den Sie zu einer kostengünstigeren Speicherklasse migrieren können. Sie können auch Metriken zur Anzahl von Lebenszyklusregeln verwenden, um Buckets zu identifizieren, in denen bestimmte Arten von Lebenszyklusregeln fehlen, z. B. Ablauf- oder Übergangsregeln.

Voraussetzung

Wenn Sie in Ihrem S3-Storage-Lens-Dashboard die Metrik Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln) sehen möchten, müssen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und

dann Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken) auswählen. Weitere Informationen finden Sie unter [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#).

Schritt 1: Identifizieren von Buckets ohne Lebenszyklusregeln

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wenn Sie bestimmte Buckets ohne Lebenszyklusregeln identifizieren möchten, scrollen Sie nach unten zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum).

Standardmäßig werden im Abschnitt Top N overview for date (Top-N-Übersicht für Datum) Metriken für die drei wichtigsten Buckets angezeigt. Im Feld Top N können Sie die Anzahl der Buckets erhöhen. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

Note

Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter [Metrikauswahl](#).

5. Wählen Sie für Metric (Metrik) in der Kategorie Cost optimization (Kostenoptimierung) die Option Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln) aus.
6. Überprüfen Sie die folgenden Daten für Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln):
 - Top number accounts (Top N-Konten) – Sehen Sie sich an, welche Konten die meisten Buckets ohne Lebenszyklusregeln haben.
 - Top number Regions (Top N-Regionen) – Sehen Sie sich eine Aufschlüsselung der Buckets ohne Lebenszyklusregeln nach Region an.
 - Top number buckets (Top N-Buckets) – Finden Sie heraus, für welche Buckets keine Lebenszyklusregeln festgelegt sind.


In jedem Diagramm oder jeder Visualisierung Ihres S3-Storage-Lens-Dashboards können Sie mithilfe der Registerkarten Account (Konto), AWS-Region, Storage classe (Speicherklasse) oder Bucket tiefere Aggregationsebenen aufrufen. Ein Beispiel finden Sie unter [Entdecken Sie kalte Amazon S3-Buckets](#).

Nachdem Sie ermittelt haben, für welche Buckets keine Lebenszyklusregeln festgelegt sind, können Sie auch die Anzahl der spezifischen Lebenszyklusregeln für Ihre Buckets überprüfen.

Schritt 2: Überprüfen der Anzahl der Lebenszyklusregeln für Ihre Buckets

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem S3-Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Cost optimization (Kostenoptimierung) für die angezeigten Buckets.

6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Kostenoptimierungsmetriken angezeigt werden, wählen Sie das Präferenzensymbol  aus.)
7. Deaktivieren Sie die Schalter für alle Metriken zur Kostenoptimierung, bis nur noch die folgenden Metriken ausgewählt sind:
 - Transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang)
 - Expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf)
 - Noncurrent version transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang nicht aktueller Versionen)
 - Noncurrent version expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf nicht aktueller Versionen)

- Abort incomplete multipart upload lifecycle rule count (Anzahl der Lebenszyklusregeln zum Abbrechen unvollständiger mehrteiliger Uploads)
 - Total lifecycle rule count (Anzahl der gesamten Lebenszyklusregeln)
8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
 9. Wählen Sie Bestätigen aus.

Die Liste Buckets wird aktualisiert und zeigt die die Metriken für die Anzahl der Lebenszyklusregeln für Ihre Buckets an. Sie können diese Daten verwenden, um Buckets ohne Lebenszyklusregeln oder Buckets zu identifizieren, für die bestimmte Arten von Lebenszyklusregeln fehlen, z. B. Ablaufregeln oder Übergangsregeln. Anschließend können Sie in der S3-Konsole zu diesen Buckets navigieren und ihnen Lebenszyklusregeln hinzufügen.

Schritt 3: Hinzufügen von Lebenszyklusregeln

Nachdem Sie Buckets ohne Lebenszyklusregeln identifiziert haben, können Sie Lebenszyklusregeln hinzufügen. Weitere Informationen finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#) und [Beispiele der S3-Lebenszyklus-Konfiguration](#).

Verwenden von S3 Storage Lens zum Schutz Ihrer Daten

Sie können die Datenschutzmetriken von Amazon S3 Storage Lens verwenden, um Buckets zu identifizieren, in denen die bewährten Datenschutzmethoden nicht angewendet wurden. Sie können diese Metriken verwenden, um Maßnahmen zu ergreifen und Standardeinstellungen anzuwenden, die den bewährten Methoden zum Schutz Ihrer Daten in allen Buckets Ihres Kontos oder Ihrer Organisation entsprechen. Sie können beispielsweise Datenschutzmetriken verwenden, um Buckets, die keine AWS Key Management Service (AWS KMS)-Schlüssel (SSE-KMS) für die Standardverschlüsselung verwenden, oder Anforderungen, die AWS Signature Version 2 (SigV2) verwenden, zu identifizieren.

Die folgenden Anwendungsfälle bieten Strategien zur Verwendung Ihres S3-Storage-Lens-Dashboards, um Ausreißer zu ermitteln und bewährte Datenschutzmethoden auf Ihre S3-Buckets anzuwenden.

Themen

- [Identifizieren von Buckets, die keine serverseitige Verschlüsselung mit AWS KMS für die Standardverschlüsselung \(SSE-KMS\) verwenden](#)

- [Identifizieren von Buckets, für die S3 Versioning aktiviert ist](#)
- [Identifizieren von Anforderungen, die AWS Signature Version 2 \(SigV2\) verwenden](#)
- [Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket](#)
- [Identifizieren des Prozentsatz der Bytes mit Objektsperre](#)

Identifizieren von Buckets, die keine serverseitige Verschlüsselung mit AWS KMS für die Standardverschlüsselung (SSE-KMS) verwenden

Mit der Amazon-S3-Standardverschlüsselung können Sie das Verhalten der Standardverschlüsselung für einen S3-Bucket festlegen. Weitere Informationen finden Sie unter [the section called "Festlegen der Standard-Bucket-Verschlüsselung"](#).

Sie können die Metriken SSE-KMS enabled bucket count (SSE-KMS-fähige Bucket-Anzahl) und % SSE-KMS enabled buckets (% SSE-KMS-fähige Buckets) verwenden, um Buckets zu identifizieren, die serverseitige Verschlüsselung mit AWS KMS-Schlüsseln (SSE-KMS) als Standardverschlüsselung verwenden. S3 Storage Lens bietet auch Metriken für unverschlüsselte Bytes, unverschlüsselte Objekte, verschlüsselte Bytes und verschlüsselte Objekte. Eine vollständige Liste der Metriken finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Sie können SSE-KMS-Verschlüsselungsmetriken im Kontext allgemeiner Verschlüsselungsmetriken analysieren, um Buckets zu identifizieren, die SSE-KMS nicht verwenden. Wenn Sie SSE-KMS für alle Buckets in Ihrem Konto oder Ihrer Organisation verwenden möchten, können Sie die Standardverschlüsselungseinstellungen für diese Buckets aktualisieren, um SSE-KMS zu verwenden. Zusätzlich zu SSE-KMS können Sie auch die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) oder vom Kunden bereitgestellten Schlüsseln (SSE-C) verwenden. Weitere Informationen finden Sie unter [Datenschutz durch Verschlüsselung](#).

Schritt 1: Identifizieren, welche Buckets SSE-KMS für die Standardverschlüsselung verwenden

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Trends and distributions (Trends und Verteilungen) die % SSE-KMS enabled bucket count (% der Anzahl der SSE-KMS-fähigen Buckets) als primäre Metrik und % encrypted bytes (% verschlüsselte Bytes) als sekundäre Metrik aus.

Das Diagramm Trend for date (Trend für Datum) wird aktualisiert und zeigt Trends für SSE-KMS und verschlüsselte Bytes an.

5. So zeigen Sie detailliertere Einblicke auf Bucket-Ebene für SSE-KMS an:
 - a. Wählen Sie einen Punkt im Diagramm aus. Es erscheint ein Feld mit Auswahlmöglichkeiten für detailliertere Einblicke.
 - b. Wählen Sie die Dimension Buckets aus. Wählen Sie dann Apply (Anwenden).
6. Wählen Sie im Diagramm Distribution by buckets for date (Verteilung nach Buckets für Datum) die Metrik SSE-KMS enabled bucket count (SSE-KMS-fähige Bucket-Anzahl) aus.
7. Sie können jetzt sehen, für welche Buckets SSE-KMS aktiviert ist und für welche nicht.

Schritt 2: Aktualisieren der Standardverschlüsselungseinstellungen für den Bucket

Nachdem Sie nun im Kontext Ihrer Metrik % encrypted bytes (% verschlüsselte Bytes) ermittelt haben, welche Buckets SSE-KMS verwenden, können Sie Buckets identifizieren, die SSE-KMS nicht verwenden. Sie können dann optional in der S3-Konsole zu diesen Buckets navigieren und deren Standardverschlüsselungseinstellungen aktualisieren, um SSE-KMS oder SSE-S3 zu verwenden. Weitere Informationen finden Sie unter [Konfigurieren der Standardverschlüsselung](#).

Identifizieren von Buckets, für die S3 Versioning aktiviert ist

Wenn die S3-Versioningfunktion aktiviert ist, behält sie mehrere Versionen desselben Objekts bei, die zur schnellen Wiederherstellung von Daten verwendet werden können, wenn ein Objekt versehentlich gelöscht oder überschrieben wird. Sie können die Metrik Versioning-enabled bucket count (Bucket-Anzahl mit aktiviertem Versioning) verwenden, um zu sehen, welche Buckets S3 Versioning verwenden. Anschließend können Sie in der S3-Konsole Maßnahmen ergreifen, um S3 Versioning für andere Buckets zu aktivieren.

Schritt 1: Identifizieren von Buckets, für die S3 Versioning aktiviert ist

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Gehen Sie im Navigationsbereich auf Storage Lens und dann auf Dashboards.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.

4. Wählen Sie im Abschnitt Trends and distributions (Trends und Verteilungen) die Metrik Versioning-enabled bucket count (Bucket-Anzahl mit aktiviertem Versioning) als primäre Metrik und Buckets als sekundäre Metrik aus.

Das Diagramm Trend for date (Trend für Datum) wird aktualisiert, um Trends für Buckets mit aktiviertem S3 Versioning anzuzeigen. Direkt unter der Trendlinie sehen Sie die Unterabschnitte Storage class distribution (Speicherlassenverteilung) und Region distribution (Regionale Verteilung).

5. Gehen Sie wie folgt vor, um detailliertere Einblicke für einen der Buckets im Diagramm Trend for date (Trend für Datum) zu sehen und eine eingehendere Analyse durchführen zu können:
 - a. Wählen Sie einen Punkt im Diagramm aus. Es erscheint ein Feld mit Auswahlmöglichkeiten für detailliertere Einblicke.
 - b. Wählen Sie eine Dimension aus, die Sie für eine eingehendere Analyse auf Ihre Daten anwenden möchten: Account (Konto), AWS-Region, Storage class (Speicherklasse) oder Bucket. Wählen Sie dann Apply (Anwenden).
6. Wählen Sie im Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) die Metriken Versioning-enabled bucket count (Bucket-Anzahl mit aktiviertem Versioning), Buckets und Active Buckets (Aktive Buckets) aus.

Der Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) wird aktualisiert und zeigt nun Daten für die ausgewählten Metriken an. Sie können diese Daten verwenden, um zu sehen, für welche Buckets S3 Versioning im Kontext Ihrer gesamten Bucket-Anzahl aktiviert ist. Im Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) können Sie Ihre Buckets in mehreren Dimensionen darstellen, indem Sie drei beliebige Metriken verwenden, um die X-axis (x-Achse), Y-axis (y-Achse) und die Size (Größe) der Blase darzustellen.

Schritt 2: Aktivieren von S3 Versioning

Nachdem Sie Buckets identifiziert haben, für die S3 Versioning aktiviert ist, können Sie Buckets identifizieren, für die S3 Versioning noch nie aktiviert war oder deren Versionsverwaltung ausgesetzt ist. Anschließend können Sie die Versionsverwaltung für diese Buckets in der S3-Konsole optional aktivieren. Weitere Informationen finden Sie unter [Aktivieren des Versioning für Buckets](#).

Identifizieren von Anforderungen, die AWS Signature Version 2 (SigV2) verwenden

Sie können die Metrik All unsupported signature requests (Alle nicht unterstützten Signaturanforderungen) verwenden, um Anfragen zu identifizieren, die AWS Signature Version 2 (SigV2) verwenden. Diese Daten können Ihnen helfen, bestimmte Anwendungen zu identifizieren, die SigV2 verwenden. Sie können diese Anwendungen dann zu AWS Signature Version 4 (SigV4) migrieren.

SigV4 ist die empfohlene Signaturmethode für alle neuen S3-Anwendungen. SigV4 bietet eine verbesserte Sicherheit und wird in allen AWS-Regionen unterstützt. Weitere Informationen finden Sie unter [Amazon S3 Update – SigV2 Deprecation Period Extended and Modified](#).

Voraussetzung


Zum Anzeigen von All unsupported signature requests (Alle nicht unterstützten Signaturanforderungen) in Ihrem S3-Storage-Lens-Dashboard müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced data protection metrics (Erweiterte Datensicherheitsmetriken) auswählen. Weitere Informationen finden Sie unter [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#).

Schritt 1: Untersuchen der SigV2-Signaturtrends nach AWS-Konto, Region und Bucket

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. So identifizieren Sie bestimmte Buckets, Konten und Regionen mit Anforderungen, die SigV2 verwenden:
 - a. Geben Sie unter Top N overview for date (Top-N-Übersicht für Datum) im Feld Top N die Anzahl der Buckets ein, deren Daten Sie sehen möchten.
 - b. Wählen Sie für Metric (Metrik) die Option All unsupported signature requests (Alle nicht unterstützten Signaturanforderungen) in der Kategorie Data protection (Datenschutz) aus.

Die Ansicht Top N overview for date (Top-N-Übersicht für Datum) wird mit den Daten für SigV2-Anforderungen nach Konto, AWS-Region und Bucket aktualisiert. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung

gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.


 Note

Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter [Metrikauswahl](#).

Schritt 2: Identifizieren von Buckets, auf die Anwendungen über SigV2-Anforderungen zugreifen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus. Löschen Sie dann Summary (Zusammenfassung).

Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Data protection (Datenschutz) für die angezeigten Buckets.

6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Datenschutzmetriken angezeigt werden, wählen Sie das Präferenzensymbol  aus.)
7. Deaktivieren Sie die Schalter für alle Datenschutzmetriken, bis nur noch die folgenden Metriken ausgewählt sind:
 - All unsupported signature requests (Alle nicht unterstützten Signaturanforderungen)
 - % all unsupported signature requests (% aller nicht unterstützten Signaturanforderungen)
8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.

9. Wählen Sie Bestätigen aus.

Die Liste Buckets wird aktualisiert, um Metriken auf Bucket-Ebene für SigV2-Anforderungen anzuzeigen. Sie können diese Daten verwenden, um bestimmte Buckets zu identifizieren, die SigV2-Anforderungen haben. Anschließend können Sie diese Informationen verwenden, um Ihre Anwendungen zu SigV4 zu migrieren. Weitere Informationen finden Sie unter [Authenticating Requests \(Authentifizierung von Anforderung\) \(AWS Signature Version 4\)](#) in der API-Referenz für Amazon Simple Storage Service.

Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket

S3 Replication ermöglicht das automatische, asynchrone Kopieren von Objekten in Amazon-S3-Buckets. Buckets, die für die Objektreplikation konfiguriert sind, können sich im Besitz desselben AWS-Konto oder unterschiedlicher Konten befinden. Weitere Informationen finden Sie unter [Überblick über das Replizieren von Objekten](#).


Sie können die Metriken zur Anzahl der Replikationsregeln von S3 Storage Lens verwenden, um detaillierte Informationen pro Bucket zu Ihren Buckets zu erhalten, die für die Replikation konfiguriert sind. Zu diesen Informationen gehören Replikationsregeln innerhalb von Buckets und Regionen sowie Bucket- und regionsübergreifende Replikationsregeln.

Voraussetzung

Zum Anzeigen von Metriken zur Anzahl der Replikationsregeln in Ihrem S3-Storage-Lens-Dashboard müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced data protection metrics (Erweiterte Datensicherheitsmetriken) auswählen. Weitere Informationen finden Sie unter [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#).

Schritt 1: Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.

5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus. Löschen Sie dann Summary (Zusammenfassung).
6. Wenn Sie die Liste Buckets so filtern möchten, dass nur Metriken zur Anzahl der Replikationsregeln angezeigt werden, wählen Sie das Präferenzensymbol  aus.
7. Deaktivieren Sie die Schalter für alle Datenschutzmetriken, bis nur noch die Metriken zur Anzahl der Replikationsregeln ausgewählt sind:
 - Same-Region Replication rule count (Anzahl der Replikationsregeln für dieselbe Region)
 - Cross-Region Replication rule count (Anzahl der regionsübergreifenden Replikationsregeln)
 - Same-account replication rule count (Anzahl der Replikationsregeln für dasselbe Konto)
 - Cross-account replication rule count (Kontenübergreifende Replikationsregelanzahl)
 - Total replication rule count (Gesamte Replikationsregelanzahl)
8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Bestätigen aus.

Schritt 2: Hinzufügen von Replikationsregeln

Nachdem Sie die Anzahl der Replikationsregeln pro Bucket festgelegt haben, können Sie optional weitere Replikationsregeln erstellen. Weitere Informationen finden Sie unter [Beispiele für die Konfiguration der Live-Replikation](#).

Identifizieren des Prozentsatz der Bytes mit Objektsperre

Mit der S3-Objektsperre können Sie Objekte anhand des Modells write-once-read-many (WORM) speichern. Mit der Objektsperre können Sie verhindern, dass Objekte für einen bestimmten Zeitraum oder auf unbestimmte Zeit gelöscht oder überschrieben werden. Sie können die Objektsperre nur aktivieren, wenn Sie einen Bucket erstellen und S3 Versioning aktivieren. Sie können jedoch den Aufbewahrungszeitraum für einzelne Objektversionen bearbeiten oder gesetzliche Aufbewahrungsfristen für Buckets anwenden, für die die Objektsperre aktiviert ist. Weitere Informationen finden Sie unter [Verwenden der S3-Objektsperre](#).

Sie können Metriken für die Objektsperre in S3 Storage Lens verwenden, um die Metrik % Object Lock bytes (% Bytes mit Objektsperre) für Ihr Konto oder Ihre Organisation anzuzeigen. Sie können diese Informationen verwenden, um Buckets in Ihrem Konto oder Ihrer Organisation zu identifizieren, die Ihre bewährten Datenschutzmethoden nicht einhalten.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus.


Der Abschnitt Snapshot wird aktualisiert und zeigt nun Datenschutzmetriken an, einschließlich der Metrik % Object Lock bytes (% Bytes mit Objektsperre). Sie können den Gesamtprozentsatz der Bytes mit Objektsperre für Ihr Konto oder Ihre Organisation sehen.

5. Wenn Sie die Metrik % Object Lock bytes (% Bytes mit Objektsperre) pro Bucket sehen möchten, scrollen Sie nach unten zum Abschnitt Top N overview (Top-N-Übersicht).

Wenn Sie Daten mit Objektsperre auf Objektebene abrufen möchten, können Sie auch die Metriken Object Lock object count (Objektsperren-Objektanzahl) und % Object Lock objects (% Objekte mit Objektsperre) verwenden.

6. Wählen Sie für Metric (Metrik) die Option % Object Lock Bytes (% Bytes mit Objektsperre) aus der Kategorie Data protection (Datenschutz) aus.

Standardmäßig werden im Abschnitt Top N overview for date (Top-N-Übersicht für Datum) Metriken für die drei wichtigsten Buckets angezeigt. Im Feld Top N können Sie die Anzahl der Buckets erhöhen. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

 Note

Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter [Metrikauswahl](#).

7. Überprüfen Sie die folgenden Daten für % Object Lock bytes (% Objekte mit Objektsperre):
- Top number accounts (Top N-Konten) – Sehen Sie sich an, welche Konten den höchsten und den niedrigsten Wert für % Object Lock bytes (% Bytes mit Objektsperre) haben.
 - Top number Regions (Top N-Regionen) – Zeigt eine Aufschlüsselung der % Object Lock bytes (% Bytes mit Objektsperre) nach Region an.
 - Top number buckets (Top N-Buckets) – Sehen Sie sich an, welche Buckets den höchsten und den niedrigsten Wert für % Object Lock bytes (% Bytes mit Objektsperre) haben.

Verwenden von S3 Storage Lens zur Überwachung der Einstellungen für die Objekteigentümerschaft

Amazon S3 Object Ownership ist eine S3-Einstellung auf Bucket-Ebene, mit der Sie Zugriffssteuerungslisten (ACLs) deaktivieren und die Eigentümerschaft der Objekte in Ihrem Bucket kontrollieren können. Wenn Sie die Einstellung für die Objekteigentümerschaft auf „Bucket owner enforced“ (Bucket-Eigentümer erzwungen) festlegen, können Sie [Zugriffssteuerungslisten \(ACLs\)](#) deaktivieren und die Eigentümerschaft für alle Objekte in Ihrem Bucket übernehmen. Dieser Ansatz vereinfacht die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten.

Wenn ein anderes AWS-Konto ein Objekt in Ihren S3-Bucket hochlädt, besitzt dieses Konto (der Objektschreiber) standardmäßig das Objekt, hat Zugriff darauf und kann anderen Benutzern über ACLs Zugriff darauf gewähren. Sie können Object Ownership verwenden, um dieses Standardverhalten zu ändern.

Die meisten modernen Anwendungsfälle in Amazon S3 erfordern keine ACLs mehr. Wir empfehlen Ihnen daher, ACLs zu deaktivieren, außer unter ungewöhnlichen Umständen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn Sie die Objekteigentümerschaft auf „Bucket owner enforced“ (Bucket-Eigentümer erzwungen) festlegen, können Sie ACLs deaktivieren und sich auf Richtlinien für die Zugriffskontrolle verlassen. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Mit den Zugriffsverwaltungsmetriken von S3 Storage Lens können Sie Buckets identifizieren, für die ACLs nicht deaktiviert sind. Nachdem Sie diese Buckets identifiziert haben, können Sie ACL-Berechtigungen zu Richtlinien migrieren und die ACLs für diese Buckets deaktivieren.

Themen

- [Schritt 1: Identifizieren allgemeiner Trends für die Einstellungen der Objekteigentümerschaft](#)
- [Schritt 2: Identifizieren von Trends auf Bucket-Ebene für die Einstellungen der Objekteigentümerschaft](#)
- [Schritt 3: Aktualisieren Ihrer Einstellung für die Objekteigentümerschaft auf „Bucket owner enforced“ \(Bucket-Eigentümer erzwungen\) zur Deaktivierung von ACLs](#)

Schritt 1: Identifizieren allgemeiner Trends für die Einstellungen der Objekteigentümerschaft

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot for date (Snapshot für Datum) unter Metrics categories (Metrikkategorien) die Option Access management (Zugriffsverwaltung) aus.

Der Abschnitt Snapshot for date (Snapshot für Datum) wird aktualisiert und zeigt nun die Metrik % Object Ownership bucket owner enforced (% Bucket-Eigentümer erzwungen für Objekteigentümerschaft) an. Sie können den Gesamtanteil an Buckets in Ihrem Konto oder Ihrer Organisation anzeigen, die die Einstellung „Bucket-Eigentümer erzwungen“ für die Objekteigentümerschaft verwenden, um ACLs zu deaktivieren.

Schritt 2: Identifizieren von Trends auf Bucket-Ebene für die Einstellungen der Objekteigentümerschaft


1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.

4. Wenn Sie detailliertere Metriken auf Bucket-Ebene anzeigen möchten, wählen Sie die Registerkarte Bucket aus.
5. Wählen Sie im Abschnitt Distribution by buckets for date (Verteilung nach Buckets für Datum) die Metrik % Object Ownership bucket owner enforced (% Bucket-Eigentümer erzwungen für Objekteigentümerschaft) aus.

Das Diagramm wird aktualisiert und zeigt nun eine Aufschlüsselung je Bucket für % Object Ownership bucket owner enforced (% Bucket-Eigentümer erzwungen für Objekteigentümerschaft) an. Sie können sehen, welche Buckets die Einstellung „Bucket-Eigentümer erzwungen“ für die Objekteigentümerschaft verwenden, um ACLs zu deaktivieren.

6. Wenn Sie die Einstellungen „Bucket-Eigentümer erzwungen“ im Kontext sehen möchten, scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Access management (Zugriffsverwaltung) aus. Löschen Sie dann Summary (Zusammenfassung).

In der Liste Buckets werden Daten für alle drei Einstellungen für die Objekteigentümerschaft angezeigt: „Bucket owner enforced“ (Bucket-Eigentümer erzwungen), „Bucket owner preferred“ (Bucket-Eigentümer bevorzugt) und „Object writer“ (Objektschreiber).

7. Wenn Sie die Liste Buckets so filtern möchten, dass Metriken nur für eine bestimmte Einstellung der Objekteigentümerschaft angezeigt werden, wählen Sie das Präferenzensymbol ).
8. Löschen Sie die Metriken, die Sie nicht anzeigen möchten.
9. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
10. Wählen Sie Confirm (Bestätigen).

Schritt 3: Aktualisieren Ihrer Einstellung für die Objekteigentümerschaft auf „Bucket owner enforced“ (Bucket-Eigentümer erzwungen) zur Deaktivierung von ACLs

Nachdem Sie Buckets identifiziert haben, die die Einstellungen „Object writer“ (Objektschreiber) und „Bucket owner preferred“ (Bucket-Eigentümer bevorzugt) für die Objekteigentümerschaft verwenden, können Sie Ihre ACL-Berechtigungen zu Bucket-Richtlinien migrieren. Wenn Sie die Migration Ihrer ACL-Berechtigungen abgeschlossen haben, können Sie Ihre Einstellungen für die Objekteigentümerschaft auf „Bucket owner enforced“ (Bucket-Eigentümer erzwungen) aktualisieren, um ACLs zu deaktivieren. Weitere Informationen finden Sie unter [Voraussetzungen für die Deaktivierung ACLs](#).

Verwenden der Metriken von S3 Storage Lens zur Verbesserung der Leistung

Wenn Sie die Option [S3 Storage Lens advanced metrics](#) (Erweiterte Metriken von S3 Storage Lens) aktiviert haben, können Sie die detaillierten Statuscode-Metriken verwenden, um die Anzahl von erfolgreichen oder fehlgeschlagenen Anforderungen zu erhalten. Sie können diese Informationen verwenden, um Probleme mit dem Zugriff oder der Leistung zu beheben. Detaillierte Statuscodemetriken zeigen die Anzahl der HTTP-Statuscodes an, z. B. 403 Forbidden und 503 Service Unavailable. Sie können allgemeine Trends anhand detaillierter Statuscode-Metriken für S3-Buckets, Konten und Organisationen untersuchen. Anschließend können Sie sich Metriken auf Bucket-Ebene genauer ansehen, um Workloads zu identifizieren, die derzeit auf diese Buckets zugreifen und Fehler verursachen.

Sie können sich beispielsweise die Metrik 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) ansehen, um Workloads zu identifizieren, die auf Buckets zugreifen, ohne dass die korrekten Berechtigungen angewendet wurden. Nachdem Sie diese Workloads identifiziert haben, können Sie außerhalb von S3 Storage Lens detaillierte Einblicke erhalten, um Ihre „403 Forbidden“-Fehler zu beheben.

Dieses Beispiel zeigt Ihnen, wie Sie eine Trendanalyse für den Fehler „403 Forbidden“ durchführen, indem Sie die Metriken 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) und % 403 Forbidden errors (% Fehler 403 Forbidden) verwenden. Anhand dieser Metriken können Sie Workloads identifizieren, die auf Buckets zugreifen, ohne dass die korrekten Berechtigungen angewendet wurden. Eine ähnliche Trendanalyse können Sie für jede der anderen Detailed status code metrics (Detaillierte Statuscode-Metriken) durchführen. Weitere Informationen finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Voraussetzung

Wenn Sie Detailed status code metrics (Detaillierte Statuscode-Metriken) in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten, müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Detailed status code metrics (Detaillierte Statuscode-Metriken) auswählen. Weitere Informationen finden Sie unter [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#).

Themen

- [Schritt 1: Durchführen einer Trendanalyse für einen einzelnen HTTP-Statuscode](#)
- [Schritt 2: Analysieren der Fehleranzahl nach Bucket](#)
- [Schritt 3: Fehlerbehebung](#)

Schritt 1: Durchführen einer Trendanalyse für einen einzelnen HTTP-Statuscode


1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Trends and distributions (Trends und Verteilungen) für Primary metric (Primäre Metrik) in der Kategorie Detailed status codes (Detaillierte Statuscodes) die Option 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) aus. Wählen Sie für Secondary metric (Sekundäre Metrik) die Option % 403 Forbidden errors (% Fehler 403 Forbidden) aus.
5. Scrollen Sie nach unten zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum). Wählen Sie für Metrics (Metriken) die Option 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) oder % 403 Forbidden errors (% Fehler 403 Forbidden) in der Kategorie Detailed status codes (Detaillierte Statuscodes) aus.

Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) wird aktualisiert und zeigt nun die Anzahl der häufigsten Fehler „403 Forbidden“ nach Konto, AWS-Region und Bucket an.

Schritt 2: Analysieren der Fehleranzahl nach Bucket

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Detailed status code metrics (Detaillierte Statuscode-Metriken) aus. Löschen Sie dann Summary (Zusammenfassung).

Die Liste Buckets wird aktualisiert und zeigt alle verfügbaren detaillierten Statuscode-Metriken an. Sie können diese Informationen verwenden, um zu sehen, welche Buckets einen großen Anteil bestimmter HTTP-Statuscodes enthalten und welche Statuscodes für die Buckets gängig sind.

6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte detaillierte Statuscode-Metriken angezeigt werden, wählen Sie das Präferenzensymbol ).
7. Deaktivieren Sie die Schalter für alle detaillierten Statuscode-Metriken, die Sie nicht in der Liste Buckets anzeigen möchten.
8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Confirm (Bestätigen).

In der Liste Buckets werden Metriken zur Fehleranzahl für die von Ihnen angegebene Anzahl von Buckets angezeigt. Sie können diese Informationen verwenden, um bestimmte Buckets zu identifizieren, in denen viele Fehler auftreten, und um Fehler nach Bucket zu beheben.

Schritt 3: Fehlerbehebung

Nachdem Sie Buckets mit einem hohen Anteil an bestimmten HTTP-Statuscodes identifiziert haben, können Sie diese Fehler beheben. Weitere Informationen finden Sie unter:

- [Warum erhalte ich die Fehlermeldung „403 Forbidden“, wenn ich versuche, Dateien in Amazon S3 hochzuladen?](#)
- [Warum erhalte ich die Fehlermeldung „403 Forbidden“, wenn ich versuche, eine Bucket-Richtlinie in Amazon S3 zu ändern?](#)
- [Wie behebe ich „403 Forbidden“-Fehler in meinem Amazon-S3-Bucket, bei dem alle Ressourcen aus demselben AWS-Konto stammen?](#)
- [Wie behebe ich einen HTTP 500- oder 503-Fehler von Amazon S3?](#)

Amazon S3-Storage-Lens-Metrik glossar

Das Metrik glossar von Amazon S3 Storage Lens enthält eine vollständige Liste der kostenlosen und erweiterten Metriken für S3 Storage Lens.

S3 Storage Lens bietet kostenlose Metriken für alle Dashboards und Konfigurationen mit der Option zum Upgrade auf erweiterte Metriken.

- Kostenlose Metriken enthalten relevante Metriken für Ihre Speichernutzung, z. B. die Anzahl der Buckets und der Objekte in Ihrem Konto. Zu den kostenlosen Metriken gehören auch

auf Anwendungsfällen basierende Metriken, etwa Metriken zur Kostenoptimierung und zum Datenschutz. Alle kostenlosen Metriken werden täglich erfasst und Daten stehen für Abfragen bis zu 14 Tage lang Daten zur Verfügung.

- Zu den erweiterten Metriken und Empfehlungen gehören neben den kostenlosen Metriken weitere Metriken, wie erweiterte Metriken zum Datenschutz und zur Kostenoptimierung. Zu den erweiterten Metriken gehören auch zusätzliche Metrikkategorien wie Aktivitätsmetriken und detaillierte Statuscode-Metriken. Erweiterte Metrikdaten stehen 15 Monate für Abfragen zur Verfügung.

Für die Verwendung von S3 Storage Lens mit fortschrittlichen Metriken und Empfehlungen fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#). Weitere Informationen über erweiterte Metriken und Empfehlungen finden Sie unter [Metrikauswahl](#).

Note

Für Storage-Lens-Gruppen sind nur Speichermetriken des kostenlosen Kontingents verfügbar. Metriken für die Advanced-Stufe sind auf Storage-Lens-Gruppenebene nicht verfügbar.

Metriknamen

Die Spalte Metrikname in der folgenden Tabelle enthält den Namen jeder S3 Storage Lens in der S3-Konsole. Die Spalte CloudWatch und Export enthält den Namen jeder Metrik in Amazon CloudWatch und die Metrik-Exportdatei, die Sie in Ihrem S3 Storage Lens-Dashboard konfigurieren können.

Abgeleitete Metrikformeln

Abgeleitete Metriken sind für den Metrikexport und die CloudWatch Veröffentlichungsoption nicht verfügbar. Sie können jedoch die in der Spalte abgeleitete Metrikformeln angegebenen Metrikformeln verwenden, um sie zu berechnen.

Interpretieren der Präfix-Symbole von Amazon S3 Storage Lens für Vielfache von Metrikeinheiten (K, M, G usw.)

Metrikeinheiten von S3 Storage Lens werden mit mehreren Präfix-Symbolen geschrieben. Diese Präfixsymbole entsprechen den Symbolen des Internationalen Einheitensystems (SI), die vom Internationalen Büro für Maße und Gewichte (BIPM) standardisiert wurden. Diese Symbole werden auch im Einheitlichen Code für Maßeinheiten (UCUM) verwendet. Weitere Informationen finden Sie unter [Liste der SI-Präfix-Symbole](#).

 Note

- Die Maßeinheit für S3-Speicherbytes ist in binären Gigabyte (GB) angegeben, wobei 1 GB 2^{30} Byte, 1 TB 2^{40} Byte und 1 PB 2^{50} Byte entspricht. Diese Maßeinheit wird auch als Gibibyte (GiB) bezeichnet, wie von der Internationalen Elektrotechnischen Kommission (IEC) definiert.
- Wenn ein Objekt basierend auf seiner Lebenszykluskonfiguration das Ende seiner Lebensdauer erreicht hat, stellt Amazon S3 das Objekt zum Entfernen in eine Warteschlange und entfernt es asynchron. Daher kann es eine Verzögerung zwischen dem Ablaufdatum und dem Datum geben, an dem Amazon S3 ein Objekt entfernt. S3 Storage Lens enthält keine Metriken für Objekte, die abgelaufen sind, aber nicht entfernt wurden. Weitere Informationen über die Ablaufaktionen im S3-Lebenszyklus finden Sie unter [Auslaufende Objekte](#).

Metrik glossar von S3 Storage Lens

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metriken
Gesamtspeicher	StorageBytes	Der Gesamtspeicher, einschließlich unvollständiger mehrteiliger Uploads, Objektmetadaten und Löschmarkierungen	Kosten	Über	N	-
Anzahl Objekte	ObjectCount	Die Gesamtzahl der Objekte	Kosten	Über	N	-
Durchschnittliche Objektgröße	-	Die durchschnittliche Objektgröße	Kosten	Über	Y	Summe (StorageBytes) /

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
						Summe (ObjectCount
Aktive Buckets	-	Die Gesamtzahl der Buckets mit Speicher > 0 Byte	Kosten	Über	Y	-
Buckets	-	Die Gesamtanzahl der Buckets	Kosten	Über	Y	-
Konten	-	Die Anzahl der Konten, deren Speicher zum Bereich gehören	Kosten	Über	Y	-
Aktuelle Versions-Bytes	CurrentVersionStorageBytes	Die Anzahl der Bytes, die eine aktuelle Version eines Objekts sind	Kosten	Kosten	N	-
% Bytes der aktuellen Version	-	Der Prozentsatz der Bytes im Bereich, die eine aktuelle Version sind	Kosten	Kosten	Y	sum (CurrentVersionStorageBytes) / sum (StorageBytes

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Anzahl der Objekte der aktuellen Version	CurrentVersionObjectCount	Die Anzahl der aktuellen Versionsobjekte	Kostenintensiv	Kostenintensiv	Nein	-
% Objekte der aktuellen Version	-	Der Prozentsatz der Objekte im Bereich, die eine aktuelle Version sind	Kostenintensiv	Kostenintensiv	Ja	summe (CurrentVersionObjectCount) / summe (ObjectCount)
Nicht aktuelle Versions-Bytes	NonCurrentVersionStorageBytes	Die Anzahl der Bytes der nicht aktuellen Version	Kostenintensiv	Kostenintensiv	Nein	-
% Bytes der nicht aktuellen Version	-	Der Prozentsatz der Bytes im Bereich, die nicht aktuelle Versionen sind	Kostenintensiv	Kostenintensiv	Ja	summe (NonCurrentVersionStorageBytes) / summe (StorageBytes)
Anzahl nicht aktueller Versionenobjekte	NonCurrentVersionObjectCount	Die Anzahl der Objekte der nicht aktuellen Versionen	Kostenintensiv	Kostenintensiv	Nein	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% Objekte der nicht aktuellen Version	-	Der Prozentsatz der Objekte im Bereich, die eine nicht aktuelle Version sind	Kosten	Kosten	Y	summe (NonCurrentVersionObjectCount) / summe () ObjectCount
Löschmarkierungsbytes	DeleteMarkerStorageBytes	Die Anzahl der Bytes im Bereich, die Löschmarkierungen sind	Kosten	Kosten	N	-
% Löschmarkierungsbytes	-	Der Prozentsatz der Bytes im Bereich, die Löschmarkierungen sind	Kosten	Kosten	Y	summe (DeleteMarkerStorageBytes) / summe () StorageBytes
Anzahl der Löschmarkierungsobjekte	DeleteMarkerObjectCount	Die Gesamtzahl der Objekte mit einer Löschmarkierung	Kosten	Kosten	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Alte Metrik	Abgeleitete Metrik
% Objekte mit Löschmarkierung	-	Der Prozentsatz der Objekte im Bereich mit einer Löschmarkierung	Kosten	Kosten	Y	summe (DeleteMarkerObjectCount) / summe (ObjectCount)
Unvollständige Bytes für mehrteilige Uploads	IncompleteMultiPartUploadStorageBytes	Die Gesamtzahl der Bytes im Bereich für unvollständige mehrteilige Uploads	Kosten	Kosten	N	-
% unvollständige Bytes für mehrteilige Uploads	-	Der Prozentsatz der Bytes im Bereich, die das Ergebnis unvollständiger mehrteiliger Uploads sind	Kosten	Kosten	Y	summe (IncompleteMultiPartUploadStorageBytes) / summe (StorageBytes)

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Anzahl unvollständiger mehrteiliger Uploads	IncompleteMultipartUploadObjectCount	Die Anzahl der Objekte im Bereich, die unvollständige mehrteilige Uploads sind	Kostenintensiv	Kostenintensiv	Nein	-
% unvollständige mehrteilige Uploads	-	Der Prozentsatz der Objekte im Bereich, die unvollständige mehrteilige Uploads sind	Kostenintensiv	Kostenintensiv	Ja	summe (IncompleteMultipartUploadObjectCount) / summe (ObjectCount)
Unvollständige mehrteilige Upload-Speicherbytes, die älter als 7 Tage sind	In7DaysOlderThanMPUStorageBytes	Die Gesamtzahl der Bytes im Bereich für unvollständige mehrteilige Uploads, die älter als 7 Tage sind	Kostenintensiv	Kostenintensiv	Nein	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kate 2	Al t	Abge te Metri kel	
% unvollständige mehrteilige Upload-Speicherbytes, die älter als 7 Tage sind	-	Die Gesamtzahl der Bytes für unvollständige mehrteilige Uploads, die älter als 7 Tage sind	Kosten i	Kosten imier	Y	Summe (IncompleteMPUStorageBytesOlderThan7Tage) / Summe() StorageBytes	
Anzahl der unvollständigen mehrteiligen Upload-Objekte, die älter als 7 Tage sind	IncompleteMPUObjectCountOlderThan	Die Anzahl der Objekte, die unvollständige mehrteilige Uploads und älter als 7 Tage sind	Kosten i	Kosten imier	N	-	

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% der Anzahl der unvollständigen mehrteiligen Upload-Objekte, die älter als 7 Tage sind	-	Die Prozentsatz der Objekte, die unvollständige mehrteilige Uploads und älter als 7 Tage sind	Kosten	Kosten	Y	Summe (IncompleteMultipartUploadCountOlderThan7Days) / Summe (ObjectCount)
Anzahl der Lebenszyklusregeln für den Übergang	TransitionLifecycleRuleCount	Die Anzahl der Lebenszyklusregeln für die Übertragung von Objekten in eine andere Speicherkategorie	Advanced	Kosten	Nicht	-
Durchschnittliche Lebenszyklusregeln für den Übergang pro Bucket	-	Die durchschnittliche Anzahl der Lebenszyklusregeln für die Übertragung von Objekten in eine andere Speicherkategorie	Advanced	Kosten	Y	Summe (TransitionLifecycleRuleCount) / Summe (DistinctNumberOfBuckets)

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitete Metrik	Abgeleitete Metrik
Anzahl der Lebenszyklusregeln für den Ablauf	ExpirationLifecycleRuleCount	Die Anzahl der Lebenszyklusregeln für das Ablaufen von Objekten	Advanced	Kosten optimiert	Nein	-
Durchschnittliche Lebenszyklusregeln für den Ablauf pro Bucket	-	Die durchschnittliche Anzahl der Lebenszyklusregeln für das Ablaufen von Objekten	Advanced	Kosten optimiert	Ja	summe (ExpirationLifecycleRuleCount) / summe () DistinctNumberOfBuckets
Anzahl der Lebenszyklusregeln für den Übergang nicht aktueller Versionen	NoncurrentVersionTransitionLifecycleRuleCount	Die Anzahl der Lebenszyklusregeln für die Übertragung von nicht aktuellen Objektversionen in eine andere Speicherklasse	Advanced	Kosten optimiert	Nein	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgete Metrikel	
Durchschnittliche Lebenszyklusregeln für den Übergang von nicht aktuellen Objektversionen pro Bucket	-	Die durchschnittliche Anzahl der Lebenszyklusregeln für die Übertragung von nicht aktuellen Objektversionen in eine andere Speicherklasse	Advanced	Kostenintensiv	Yes	summe (NoncurrentVersionTransitionLifecycleRuleCount) / summe () DistinctNumberOfBuckets
Anzahl der Lebenszyklusregeln für den Ablauf nicht aktueller Versionen	NoncurrentVersionExpirationLifecycleRuleCount	Die Anzahl der Lebenszyklusregeln für das Ablaufen von nicht aktuellen Objektversionen	Advanced	Kostenintensiv	No	-
Durchschnittliche Lebenszyklusregeln für den Ablauf von nicht aktuellen Objektversionen pro Bucket	-	Die durchschnittliche Anzahl der Lebenszyklusregeln für das Ablaufen von nicht aktuellen Objektversionen	Advanced	Kostenintensiv	Yes	summe (NoncurrentVersionExpirationLifecycleRuleCount) / summe () DistinctNumberOfBuckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kate 2	Al t	Abge te Metri kel	
Anzahl der Lebenszyklusregeln zum Abbrechen unvollständiger mehrteiliger Uploads	AbortIncompleteMPULifecycleRuleCount	Die Anzahl der Lebenszyklusregeln zum Löschen unvollständiger mehrteiliger Uploads	Advanced	Kostenintensiv	Nicht	-	
Die durchschnittliche Anzahl von Lebenszyklusregeln zum Abbruch unvollständiger mehrteiliger Uploads pro Bucket	-	Die durchschnittliche Anzahl der Lebenszyklusregeln zum Löschen unvollständiger mehrteiliger Uploads	Advanced	Kostenintensiv	Yes	summe (AbortIncompleteMPULifecycleRuleCount) / summe () DistinctNumberOfBuckets	
Anzahl der Lebenszyklusregeln für abgelaufene Objekte mit Löschmarkierung	ExpiredObjectDeleteMarkerLifecycleRuleCount	Anzahl der Lebenszyklusregeln zum Entfernen abgelaufener Löschmarkierungen für Objekte	Advanced	Kostenintensiv	Nicht	-	

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Durchschnittliche Anzahl der Lebenszyklusregeln zum Entfernen abgelaufener Löschmarkierungen für Objekte	-	Die durchschnittliche Anzahl der Lebenszyklusregeln zum Entfernen abgelaufener Löschmarkierungen für Objekte	Advanced	Kostenintensiv	Yes	summe (ExpiredObjectDeleteMarkerLifecycleRuleCount) / summe () DistinctNumberOfBuckets
Anzahl der gesamten Lebenszyklusregeln	TotalLifecycleRuleCount	Die Gesamtzahl der Lebenszyklusregeln	Advanced	Kostenintensiv	No	-
Durchschnittliche Anzahl von Lebenszyklusregeln pro Bucket	-	Die durchschnittliche Anzahl der Lebenszyklusregeln	Advanced	Kostenintensiv	Yes	summe (TotalLifecycleRuleCount) / summe () DistinctNumberOfBuckets
Verschlüsselte Bytes	EncryptedStorageBytes	Die Gesamtzahl der verschlüsselten Bytes	Kostenintensiv	Datenschutz	No	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% verschlüsselte Bytes	-	Der Prozentsatz der Gesamtzahl von Bytes, die verschlüsselt sind	Kosten	Datum	Y	summe (EncryptedObjectCount) / summe () StorageBytes
Anzahl der verschlüsselten Objekte	Encrypted ObjectCount	Die Gesamtzahl der Objekte, die verschlüsselt sind	Kosten	Datum	N	-
% verschlüsselte Objekte	-	Der Prozentsatz der Objekte, die verschlüsselt sind	Kosten	Datum	Y	summe (EncryptedStorageBytes) / summe () ObjectCount
Unverschlüsselte Bytes	UnencryptedStorage Bytes	Die Anzahl der unverschlüsselten Bytes	Kosten	Datum	Y	summe (StorageBytes) - Summe () EncryptedStorageBytes

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% unverschlüsselte Bytes	-	Der Prozentsatz der Bytes, die unverschlüsselt sind	Kosten	Datentyp	Y	Summe (UnencryptedStorageBytes) / Summe (StorageBytes)
Unverschlüsselte Objektanzahl	UnencryptedObjectCount	Die Gesamtzahl der Objekte, die unverschlüsselt sind	Kosten	Datentyp	Y	summe (ObjectCount) - Summe (EncryptedObjectCount)
% unverschlüsselte Objekte	-	Der Prozentsatz der unverschlüsselten Objekte	Kosten	Datentyp	Y	Summe (UnencryptedStorageBytes) / Summe (ObjectCount)
Replizierte Speicher-Bytes-Quelle	ReplicatedStorageBytesSource	Die Gesamtzahl der Bytes, die aus dem Quell-Bucket repliziert werden	Kosten	Datentyp	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% replizierte Bytes-Quelle	-	Der Prozentsatz der Gesamtzahl der Bytes, die aus dem Quell-Bucket repliziert werden	Kosten	Daten	Y	summe (ReplicatedStorageBytesSource) / summe () StorageBytes
Quelle der Anzahl der replizierten Objekte	ReplicatedObjectCountSource	Die Anzahl der replizierten Objekte aus dem Quell-Bucket	Kosten	Daten	N	-
% replizierte Objektquelle	-	Der Prozentsatz der Gesamtzahl der Objekte, die aus dem Quell-Bucket repliziert werden	Kosten	Daten	Y	summe (ReplicatedStorageObjectCount) / summe () ObjectCount
Ziel der Replikationsspeicher-Bytes	ReplicatedStorageBytes	Die Gesamtzahl der Bytes, die in den Ziel-Bucket repliziert werden	Kosten	Daten	Y	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Alte Metrik	Abgeleitete Metrik
% repliziertes Bytes-Ziel	-	Der Prozentsatz der Gesamtzahl der Bytes, die in den Ziel-Bucket repliziert werden	Kosten	Datensatz	Y	summe (ReplicatedStorageBytes) / summe (StorageBytes)
Ziel der Anzahl der replizierten Objekte	ReplicatedObjectCount	Die Anzahl der Objekte, die in den Ziel-Bucket repliziert werden	Kosten	Datensatz	Y	-
% Ziel replizierter Objekte	-	Der Prozentsatz der Gesamtzahl der Objekte, die in den Ziel-Bucket repliziert werden	Kosten	Datensatz	Y	summe (ReplicatedObjectCount) / summe (ObjectCount)

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Bytes mit Objektsperre	ObjectLockEnabledStorageBytes	Die Gesamtzahl der Speicherbytes mit aktivierter Objektsperre	Kosten	Datentz	Y	summe (UnencryptedStorageBytes) / summe (ObjectLockEnabledStorageCount) - summe () ObjectLockEnabledStorageBytes
% Bytes mit Objektsperre	-	Der Prozentsatz der Speicherbytes mit aktivierter Objektsperre	Kosten	Datentz	Y	Summe (ObjectLockEnabledStorageBytes) / Summe () StorageBytes
Anzahl der Objekte mit aktivierter Objektsperre	ObjectLockEnabledObjectCount	Die Gesamtzahl der Objekte mit Objektsperre	Kosten	Datentz	Y	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% Objekte mit Objektsperre	-	Der Prozentsatz der Gesamtzahl der Objekte mit aktivierter Objektsperre	Kosten	Datentz	Y	summe (ObjectLockEnabledObjectCount) / summe (ObjectCount)
Bucket-Anzahl mit aktiviertem Versioning	VersioningEnabledBucketCount	Die Anzahl der Buckets, für die S3 Versioning aktiviert ist	Kosten	Datentz	N	-
% Buckets mit aktiviertem Versioning	-	Der Prozentsatz der Buckets, für die S3 Versioning aktiviert ist	Kosten	Datentz	Y	summe (VersioningEnabledBucketCount) / summe (DistinctNumberOfBuckets)
MFAAnzahl der Buckets mit aktivierter Löschfunktion	MFADeleteEnabledBucketCount	Die Anzahl der Buckets, für die das Löschen MFA (Multi-Faktor-Authentifizierung) aktiviert ist	Kosten	Datentz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kate 2	Al t	Abge te Metri kel
% MFA der Buckets, für die das Löschen aktiviert wurde	-	Der Prozentsatz der Buckets, für die das Löschen MFA (Multi-Faktor-Authentifizierung) aktiviert ist	Kost i	Date tz	Y	Summe (MFADeleteEnabledBucketCount) / Summe () DistinctNumberOfBuckets
SSE- Anzahl der Buckets KMS aktiviert	SSEKMSEnabledBucketCount	Die Anzahl der Buckets, die serverseitige Verschlüsselung mit AWS Key Management Service Schlüsseln (SSE-KMS) für die Standard-Bucket-Verschlüsselung verwenden	Kost i	Date tz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kate 2	Al t	Abge te Metri kel	
% SSE — aktivierte Buckets KMS	-	Der Prozentsatz der Buckets, KMS für die SSE Standard-Bucket-Verschlüsselung	Kosten	Datensatz	Y	sum	(SSEKMSEnableBucketCount) / sum() DistinctNumberOfBuckets
Alle nicht unterstützten Signaturanforderungen	AllUnsupportedSignatureRequests	Die Gesamtzahl der Anfragen, die nicht unterstützte Signaturversionen AWS verwenden	Advanced	Datensatz	N	-	
% aller nicht unterstützten Signaturanforderungen	-	Der Prozentsatz der Anfragen, die nicht unterstützte Signaturversionen AWS verwenden	Advanced	Datensatz	Y	sum	(AllUnsupportedSignatureRequests) / sum() AllRequests

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Alle nicht unterstützten Anfragen TLS	AllUnsupportedTLSRequests	Die Anzahl der Anfragen, die nicht unterstützte Transport Layer Security (TLS) -Versionen verwenden	Advanced	Datenzeit	Nein	-
% aller nicht unterstützten Anfragen TLS	-	Der Prozentsatz der Anfragen, die nicht unterstützte TLS Versionen verwenden	Advanced	Datenzeit	Ja	$\frac{\text{sum}(\text{AllUnsupportedTLSRequests})}{\text{sum}(\text{AllRequests})}$
Alle SSE - Anfragen KMS	AllSSEKMSRequests	Die Gesamtzahl der Anfragen, die SSE - angeben KMS	Advanced	Datenzeit	Nein	-
% aller SSE — KMS Anfragen	-	Der Prozentsatz der Anfragen, die SSE - angeben KMS	Advanced	Datenzeit	Ja	$\frac{\text{Summe}(\text{AllSSEKMSRequests})}{\text{Summe}(\text{AllRequests})}$

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Anzahl der Replikationsregeln für dieselbe Region	SameRegionReplicationRuleCount	Die Anzahl der Replikationsregeln für die Replikation in derselben Region () SRR	Advanced	Datenzeit	Nein	-
Durchschnittliche Replikationsregeln für Replikation innerhalb derselben Region	-	Die durchschnittliche Anzahl von Replikationsregeln für SRR	Advanced	Datenzeit	Ja	sum (SameRegionReplicationRuleCount) / sum () DistinctNumberOfBuckets
Anzahl der regionsübergreifenden Replikationsregeln	CrossRegionReplicationRuleCount	Die Anzahl der Replikationsregeln für die regionsübergreifende Replikation () CRR	Advanced	Datenzeit	Nein	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kate 2	Al t	Abge te Metri kel
Durchschnittliche Replikationsregeln für regionsübergreifende Replikation pro Bucket	-	Die durchschnittliche Anzahl von Replikationsregeln für CRR	Advanced	Daily	Yes	sum(CrossRegionReplicationRuleCount) / sum(DistinctNumberOfBuckets)
Anzahl der Replikationsregeln für dasselbe Konto	SameAccountReplicationRuleCount	Die Anzahl der Replikationsregeln für die Replikation innerhalb desselben Kontos	Advanced	Daily	No	-
Durchschnittliche Replikationsregeln für Replikation innerhalb desselben Kontos pro Bucket	-	Die durchschnittliche Anzahl der Replikationsregeln für die Replikation innerhalb desselben Kontos	Advanced	Daily	Yes	summe(SameAccountReplicationRuleCount) / summe(DistinctNumberOfBuckets)

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Kontenübergreifende Replikationsregelanzahl	CrossAccountReplicationRuleCount	Die Anzahl der Replikationsregeln für die kontoübergreifende Replikation	Advanced	Datenzeit	Nein	-
Durchschnittliche Replikationsregeln für die kontoübergreifende Replikation pro Bucket	-	Die durchschnittliche Anzahl der Replikationsregeln für die kontoübergreifende Replikation	Advanced	Datenzeit	Ja	summe (CrossAccountReplicationRuleCount) / summe () DistinctNumberOfBuckets
Ungültige Anzahl der Zielreplikationsregeln	InvalidDestinationReplicationRuleCount	Die Anzahl der Replikationsregeln mit einem Replikationsziel, das nicht gültig ist	Advanced	Datenzeit	Nein	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrikformel
Durchschnittliche ungültige Anzahl der Zielreplikationsregeln	-	Die durchschnittliche Anzahl der Replikationsregeln mit einem Replikationsziel, das nicht gültig ist	Advanced	Datenzeit	Y	$\frac{\text{summe (InvalidReplicationRuleCount)}}{\text{summe () DistinctNumberOfBuckets}}$
Gesamte Replikationsregelnanzahl	-	Die Gesamtzahl der Replikationsregeln	Advanced	Datenzeit	Y	-
Durchschnittliche Anzahl der Replikationsregeln pro Bucket	-	Die durchschnittliche Gesamtzahl der Replikationsregeln	Advanced	Datenzeit	Y	$\frac{\text{sum (alle Metriken zur Anzahl der Replikationsregeln)}}{\text{sum () DistinctNumberOfBuckets}}$

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Anzahl der Buckets mit Bucket-Eigentümergezwungen für Objekteigentümerschaft	ObjectOwnershipBucketOwnerEnforcedBucketCount	Die Gesamtzahl der Buckets, bei denen die Zugriffskontrolllisten (ACLs) mithilfe der vom Bucket-Besitzer erzwungenen Einstellung für Object Ownership deaktiviert wurden	Kosten	Zugriffserweiterung	Nein	-
% Bucket-Eigentümergezwungen für Objekteigentümerschaft	-	Der Prozentsatz der Buckets, die mithilfe der vom Bucket-Besitzer erzwungenen Einstellung für Objekteigentum ACLs deaktiviert wurden	Kosten	Zugriffserweiterung	Ja	Summe (ObjectOwnershipBucketOwnerEnforcedBucketCount) / Summe (DistinctNumberOfBuckets)

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
Anzahl der Buckets mit Bucket-Eigentümer bevorzugt für Objekteigentümerschaft	ObjectOwnershipBucketOwnerPreferredBucketCount	Die Gesamtzahl der Buckets, die die Einstellung „Bucket-Eigentümer bevorzugt“ für die Objekteigentümerschaft verwenden	Kosten	Zugriff	Nein	-
% der Buckets mit Bucket-Eigentümer bevorzugt für Objekteigentümerschaft	-	Der Prozentsatz der Buckets, die die Einstellung „Bucket-Eigentümer bevorzugt“ für die Objekteigentümerschaft verwenden	Kosten	Zugriff	Ja	summe (ObjectOwnershipBucketOwnerPreferredBucketCount) / summe (DistinctNumberOfBuckets)
Objekt-Autor Bucket-Anzahl des Objekt-Eigentümers	ObjectOwnershipObjectWriterBucketCount	Die Gesamtzahl der Buckets, die die Einstellung „Objektschreiber“ für die Objekteigentümerschaft verwenden	Kosten	Zugriff	Nein	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% der Buckets mit Objektschreiber für Objekteigentümerschaft	-	Der Prozentsatz der Buckets, die die Einstellung „Objektschreiber“ für die Objekteigentümerschaft verwenden	Kosten	Zugriff	Y	summe (ObjectOwnershipObjectWriterBucketCount) / summe () DistinctNumberOfBuckets
Anzahl aktivierter Buckets für Transfer Acceleration	TransferAccelerationEnabledBucketCount	Die Gesamtzahl der Buckets, für die Transfer Acceleration aktiviert ist	Kosten	Leist.	N	-
% der aktivierten Buckets für Transfer Acceleration	-	Der Prozentsatz der Buckets, für die Transfer Acceleration aktiviert ist	Kosten	Leist.	Y	summe (TransferAccelerationEnabledBucketCount) / summe () DistinctNumberOfBuckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Attribut	Abgetragene Metriken
Anzahl der aktivierten Buckets für Ereignisbenachrichtigungen	EventNotificationEnabledBucketCount	Die Gesamtzahl der Buckets, für die Ereignisbenachrichtigungen aktiviert sind	Kostenlos	Ereignis	Nicht	
% der aktivierten Buckets für Ereignisbenachrichtigungen	-	Der Prozentsatz der Buckets, für die Ereignisbenachrichtigungen aktiviert sind	Kostenlos	Ereignis	Y	summe (EventNotificationEnabledBucketCount) / summe (DistinctNumberOfBuckets)
Alle Anfragen	AllRequests	Die Gesamtzahl der gesendeten - Anfragen	Advanced	Aktiv	Nicht	-
GET-Anfragen	GetRequests	Die Gesamtzahl der gesendeten GET-Anfragen	Advanced	Aktiv	Nicht	-
PUT-Anfragen	PutRequests	Die Gesamtzahl der gesendeten PUT-Anfragen	Advanced	Aktiv	Nicht	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe ¹	Kate ²	Al t	Abge te Metri kel
HEAD-Anforderungen	HeadRequests	Die Gesamtzahl der gesendeten HEAD-Anforderungen	Advanced	Active	N	-
DELETE-Anforderungen	DeleteRequests	Die Gesamtzahl der gesendeten DELETE-Anforderungen	Advanced	Active	N	-
Listenanforderungen	ListRequests	Die Gesamtzahl der gesendeten LIST-Anforderungen	Advanced	Active	N	-
POST-Anforderungen	PostRequests	Die Gesamtzahl der gesendeten POST-Anforderungen	Advanced	Active	N	-
SELECT-Anforderungen	SelectRequests	Die Gesamtzahl der S3-SELECT-Anforderungen	Advanced	Active	N	-
Ausgewählte gescannte Bytes	SelectScannedBytes	Die Anzahl der gescannten S3-SELECT-Bytes	Advanced	Active	N	-
Ausgewählte zurückgegebene Bytes	SelectReturnedBytes	Die Anzahl der zurückgegebenen S3-SELECT-Bytes	Advanced	Active	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kate 2	Al t	Abge te Metri kel	
Heruntergeladene Bytes	BytesDownloaded	Die Anzahl der Bytes, die heruntergeladen wurden	Advanced	Active	N	-	
% Abruftrate	-	Der Prozentsatz der Bytes, die heruntergeladen wurden	Advanced	Active	Y	summe (BytesDownloaded) / summe () StorageBytes	
Hochgeladene Bytes	BytesUploaded	Die Anzahl der hochgeladenen Bytes	Advanced	Active	N	-	
% Aufnahmeverhältnis	-	Der Prozentsatz der Bytes, die hochgeladen wurden	Advanced	Active	Y	summe (BytesUploaded) / summe () StorageBytes	
4xx-Fehler	4xxErrors	Die Gesamtzahl der HTTP 4xx-Statuscodes	Advanced	Active	N	-	
5xx-Fehler	5xxErrors	Die Gesamtzahl der HTTP 5xx-Statuscodes	Advanced	Active	N	-	

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Alert	Abgeleitete Metriken
Gesamtfehler	-	Die Summe aller 4xx- und 5xx-Fehler	Advanced	Active	Yes	Summe (4xxErrors) + Summe (5xxErrors)
% Fehlerrate	-	Die Gesamtzahl der 4xx- und 5xx-Fehler als Prozentsatz der Gesamtanforderungen	Advanced	Active	Yes	summe (TotalErrors) / summe (TotalRequests)
200 OK Statusanzahl	200 OKStatusCount	Die Gesamtzahl der 200-OK-Statuscodes	Advanced	Detailed	No	-
% Status 200 OK	-	Die Gesamtzahl der 200-OK-Statuscodes als Prozentsatz der Gesamtzahl der Anforderungen	Advanced	Detailed	Yes	Summe (200OKStatusCount) / Summe (AllRequests)
Statusanzahl 206 unvollständige Inhalte	206 PartialContentStatusCount	Die Gesamtzahl der Statuscodes 206 für unvollständige Inhalte	Advanced	Detailed	No	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kategorie 2	Abgeleitet	Abgeleitete Metrik
% Status 206 unvollständige Inhalte	-	Die Gesamtzahl der Statuscodes 206 für unvollständige Inhalte als Prozentsatz der Gesamtzahl der Anforderungen	Advanced	Detailed	Yes	Summe (206PartialContentStatusCount) / Summe (AllRequests)
Anzahl der Fehler 400 Bad Request	400 BadRequestErrorCount	Die Gesamtzahl der Statuscodes 400 Bad Request	Advanced	Detailed	No	-
% Fehler 400 Bad Request	-	Die Gesamtzahl der Statuscodes 400 Bad Request als Prozentsatz der Gesamtzahl der Anforderungen	Advanced	Detailed	Yes	Summe (400BadRequestErrorCount) / Summe (AllRequests)
Anzahl der Fehler 403 Forbidden	403 ForbiddenErrorCount	Die Gesamtzahl der Statuscodes 403 Forbidden	Advanced	Detailed	No	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe 1	Kate 2	Al t	Abge te Metri kel	
% Fehler 403 Forbidden	-	Die Gesamtzahl der Statuscodes 403 Forbidden als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Detat	Y	Summe (403ForbiddenErrorCount) / Summe (AllRequests)	
Anzahl der Fehler 404 Not Found	404 NotFoundErrorCount	Die Gesamtzahl der Statuscodes 404 Not Found	Adva	Detat	N	-	
% Fehler 404 Not Found	-	Die Gesamtzahl der Statuscodes 404 Not Found als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Detat	Y	Summe (404NotFoundAndErrorCount) / Summe (AllRequests)	
Fehleranzahl 500 Internal Server Error	500 InternalServerErrorCount	Die Gesamtzahl der Statuscodes 500 Internal Server Error	Adva	Detat	N	-	

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe ¹	Kategorie ²	Abgeleitete Metrik	
% Fehler 500 Internal Server Error	-	Die Gesamtzahl der Statuscodes 500 Internal Server Error als Prozentsatz der Gesamtzahl der Anforderungen	Advanced	Detailed	Yes	Summe (500InternalServerErrorCount) / Summe (AllRequests)
Anzahl der Fehler 503 Service Unavailable	503 ServiceUnavailable ErrorCount	Die Gesamtzahl der Statuscodes 503 Service Unavailable	Advanced	Detailed	No	-
% Fehler 503 Service Unavailable	-	Die Gesamtzahl der Statuscodes 503 Service Unavailable als Prozentsatz der Gesamtzahl der Anforderungen	Advanced	Detailed	Yes	Summe (503ServiceUnavailableErrorCount) / Summe (AllRequests)

¹ Alle Metriken für das kostenlose Speicherkontingent sind auf Storage-Lens-Gruppenebene verfügbar. Metriken für die Advanced-Stufe sind auf Storage-Lens-Gruppenebene nicht verfügbar.

² Metriken für die Anzahl der Regeln und Metriken für Bucket-Einstellungen sind auf Präfixebene nicht verfügbar.

Arbeiten mit Amazon S3 Storage Lens unter Verwendung der Konsole und der API

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können Metriken von S3 Storage Lens verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden.

Dieser Abschnitt enthält Beispiele für das Erstellen, Aktualisieren und Anzeigen von S3-Storage-Lens-Konfigurationen und das Ausführen von funktionsbezogenen Vorgängen. Auch Anwendungsfälle für den Einsatz von S3 Storage Lens mit AWS Organizations werden in diesen Beispielen abgedeckt. Ersetzen Sie alle Variablenwerte in den Beispielen durch Ihre eigenen Werte.

Themen

- [Verwenden von Amazon S3 Storage Lens in der Konsole](#)
- [Amazon S3-Storage-Lens-Beispiele für die Verwendung der AWS CLI.](#)
- [Amazon-S3-Storage-Lens-Beispiele für die Verwendung des SDK for Java](#)

Verwenden von Amazon S3 Storage Lens in der Konsole

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können Metriken von S3 Storage Lens verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden.

Note

Es kann bis zu 48 Stunden dauern, bis Änderungen an der Dashboard-Konfiguration genau angezeigt bzw. visualisiert werden.

Themen

- [Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards](#)
- [Deaktivieren oder Löschen von Amazon S3-Storage-Lens-Dashboards](#)
- [Arbeiten mit AWS Organizations , um Dashboards auf Organisationsebene zu erstellen](#)

Erstellen und Aktualisieren von Amazon S3-Storage-Lens-Dashboards

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden.

Das Standard-Dashboard von Amazon S3 Storage Lens ist default-account-dashboard. Dieses Dashboard ist von Amazon S3 vorkonfiguriert, damit Sie zusammengefasste Erkenntnisse und Trends der aggregierten kostenlosen und erweiterten Metriken Ihres gesamten Kontos in der Konsole visualisieren können. Sie können den Konfigurationsbereich des Standard-Dashboards nicht ändern, können aber die Metrikauswahl von den kostenlosen Metriken auf die kostenpflichtigen erweiterten Metriken und Empfehlungen upgraden, den optionalen Metrikexport konfigurieren oder sogar das Standard-Dashboard deaktivieren. Das Standard-Dashboard kann nicht gelöscht werden.

Sie können auch zusätzliche benutzerdefinierte S3 Storage Lens-Dashboards erstellen, die auf Ihr Unternehmen oder auf bestimmte Regionen AWS Organizations oder Bereiche innerhalb eines Kontos zugeschnitten werden können.

Erstellen eines Amazon S3-Storage-Lens-Dashboards


Verwenden Sie die folgenden Schritte, um ein Amazon S3-Storage-Lens-Dashboard auf der Amazon S3-Konsole zu erstellen.

Schritt 1: Festlegen des Dashboard-Umfangs

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten AWS Region aus. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich unter S3 Storage Lens die Option Dashboards aus.
4. Klicken Sie auf Create dashboard (Dashboard erstellen).
5. Führen Sie auf der Seite Dashboard im Abschnitt Allgemein die folgenden Schritte aus:

- a. Sehen Sie sich die Heimatregion für Ihr Dashboard an. In der Heimatregion werden die AWS-Region Konfiguration und die Metriken für dieses Storage Lens-Dashboard gespeichert.
- b. Geben Sie einen Dashboard-Namen ein.


Dashboard-Namen müssen weniger als 65 Zeichen lang sein und dürfen keine Sonderzeichen oder Leerzeichen enthalten.

 Note

Nach dem Erstellen des Dashboards kann der Name nicht mehr geändert werden.


- c. Sie haben die Wahl, Ihrem Dashboard Markierungen hinzuzufügen. Sie können Markierungen verwenden, um Berechtigungen für Ihr Dashboard zu verwalten und die Kosten für S3 Storage Lens nachzuverfolgen.

Weitere Informationen finden Sie unter [Steuern des Zugriffs mit Ressourcentags](#) im IAM-Benutzerhandbuch und [unter AWS-generierte Kostenzuordnungstags](#) im AWS Billing - Benutzerhandbuch.

 Note

Sie können Ihrer Dashboard-Konfiguration bis zu 50 Markierungen hinzufügen.

6. Führen Sie im Abschnitt Dashboard-Bereich die folgenden Schritte aus:
 - a. Wählen Sie die Regionen und Buckets aus, die S3 Storage Lens im Dashboard ein- oder ausschließen soll.
 - b. Wählen Sie die Buckets in den ausgewählten Regionen aus, die S3 Storage Lens ein- oder ausschließen soll. Sie können Buckets entweder ein- oder ausschließen. Beides ist nicht möglich. Diese Option ist nicht verfügbar, wenn Sie Dashboards auf Organisationsebene erstellen.

 Note

- Sie können Regionen und Buckets entweder ein- oder ausschließen. Diese Option ist nur auf Regionen beschränkt, wenn Sie Dashboards auf Organisationsebene für alle Mitgliedskonten in Ihrer Organisation erstellen.
- Sie können bis zu 50 Buckets auswählen, die Sie einschließen oder ausschließen möchten.

Schritt 2: Konfigurieren der Metrikauswahl

1. Wählen Sie im Abschnitt Metrikauswahl den Metriktyp aus, den Sie für dieses Dashboard aggregieren möchten.
 - Wählen Sie Free metrics (Kostenlose Metriken) aus, um kostenlose Metriken einzuschließen, die auf Bucket-Ebene aggregiert und 14 Tage lang für Abfragen verfügbar sind.
 - Wählen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) aus, um erweiterte Metriken und andere erweiterte Optionen zu aktivieren. Zu diesen Optionen gehören erweiterte Präfixaggregation, CloudWatch Amazon-Veröffentlichung und kontextuelle Empfehlungen. Daten stehen für Abfragen für 15 Monate zur Verfügung. Erweiterte Metriken und Empfehlungen sind mit zusätzlichen Kosten verbunden. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Weitere Informationen zu erweiterten und zu kostenlosen Metriken finden Sie unter [Metrikauswahl](#).

2. Wählen Sie unter Advanced metrics and recommendations features (Erweiterte Metrik- und Empfehlungsfunktionen) die Optionen aus, Sie aktivieren möchten:
 - Advanced metrics (Erweiterte Metriken)
 - CloudWatch Veröffentlichen
 - Präfixzusammenfassung

⚠ Important

Wenn Sie die Präfixaggregation für Ihre S3 Storage Lens-Konfiguration aktivieren, werden Metriken auf Präfixebene nicht veröffentlicht. CloudWatch Nur S3 Storage Lens-Metriken auf Bucket-, Konto- und Organisationsebene werden veröffentlicht. CloudWatch

3. Wenn Sie Advanced Metrics (Erweiterte Metriken) aktiviert haben, wählen Sie die Advanced metrics categories (Erweiterte Metrikkategorien) aus, die Sie in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten:

- Metriken für Aktivitäten
- Detailed status code metrics (Detaillierte Statuscode-Metriken)
- Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken)
- Advanced data protection metrics (Erweiterte Datensicherheitsmetriken)

Weitere Informationen zu den Metrikkategorien finden Sie unter [Metrikkategorien](#). Eine vollständige Liste der Metriken finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

4. Wenn Sie die Präfixaggregation aktivieren möchten, konfigurieren Sie Folgendes:

a. Wählen Sie die Mindestgröße für den Präfixschwellenwert für dieses Dashboard aus.

Ein Präfixschwellenwert von 5 Prozent gibt beispielsweise an, dass Präfixe, die einen Anteil des gesamten Bucket-Speichers von 5 Prozent oder mehr ausmachen, aggregiert werden.

b. Wählen Sie die Präfixtiefe aus.

Diese Einstellung gibt die maximale Anzahl von Ebenen an, bis zu denen die Präfixe ausgewertet werden. Die Präfixtiefe muss weniger als 10 betragen.

c. Geben Sie ein Präfixtrennzeichen ein.

Mit diesem Wert werden die einzelnen Präfixebenen identifiziert. Der Standardwert in Amazon S3 ist das Zeichen /, in Ihrer Speicherstruktur können jedoch ggf. andere Trennzeichen verwendet werden.

(Optional) Schritt 3: Exportieren von Metriken für das Dashboard

1. Wählen Sie im Abschnitt Metrics export (Metrik-Export) die Option Enable (Aktivieren) aus, um einen Metrik-Export zu erstellen, der jeden Tag in einem Ziel-Bucket Ihrer Wahl abgelegt wird.

Der Metrik-Export liegt im CSV- oder Apache Parquet-Format vor. Er enthält den gleichen Datenumfang wie Ihr S3-Storage-Lens-Dashboard ohne die Empfehlungen.

2. Wenn Sie den Metrik-Export aktiviert haben, wählen Sie das Ausgabeformat für den täglichen Metrik-Export aus: CSV oder Apache Parquet.

Parquet ist ein Open-Source-Dateiformat für Hadoop, das verschachtelte Daten in einem flachen Spaltenformat speichert.

3. Wählen Sie den S3-Ziel-Bucket für den Export Ihrer Metriken aus.

Sie können einen Bucket im aktuellen Konto des S3-Storage-Lens-Dashboards auswählen. Sie können auch einen anderen auswählen, AWS-Konto wenn Sie über die Berechtigungen für den Ziel-Bucket und die Konto-ID des Besitzers des Ziel-Buckets verfügen.

4. Wählen Sie den S3-Ziel-Bucket (Format: `s3://bucket-name/prefix`) aus.

Der Bucket muss sich in der Heimatregion Ihres S3-Storage-Lens-Dashboards befinden. In der S3-Konsole wird das Feld Destination bucket permission (Ziel-Bucket-Berechtigung) mit der Berechtigung angezeigt, die der Ziel-Bucket-Richtlinie von Amazon S3 hinzugefügt wird. Amazon S3 aktualisiert die Bucket-Richtlinie für den Ziel-Bucket, damit S3 Daten darin ablegen kann.

5. (Optional) Wenn Sie die serverseitige Verschlüsselung für Ihren Metriken aktivieren möchten, wählen Sie Specify an encryption key (Einen Verschlüsselungsschlüssel angeben) aus. Wählen Sie anschließend den Verschlüsselungstyp aus: Von Amazon S3 verwaltete Schlüssel (SSE-S3) oder AWS Key Management Service -Schlüssel (SSE-KMS).

Sie können entweder einen [von Amazon S3 verwalteten Schlüssel](#) (SSE-S3) und einen [AWS Key Management Service \(AWS KMS\)](#)-Schlüssel (SSE-KMS) auswählen.

6. (Optional) Um einen AWS KMS Schlüssel anzugeben, müssen Sie einen KMS-Schlüssel wählen oder einen Amazon Resource Name (ARN) -Schlüssel eingeben.

Wenn Sie einen kundenverwalteten Schlüssel auswählen, müssen Sie S3 Storage Lens in der AWS KMS -Schlüsselrichtlinie die Berechtigung zum Verschlüsseln erteilen. Weitere Informationen finden Sie unter [Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln](#).

7. Klicken Sie auf Dashboard erstellen.

Um sich einen besseren Überblick über Ihren Speicher zu verschaffen, können Sie eine oder mehrere S3-Storage-Lens-Gruppen erstellen und diese an das Dashboard anhängen. Eine Storage-Lens-Gruppe ist ein benutzerdefinierter Filter für Objekte, der auf Präfixen, Suffixen, Objekt-Tags, Objektgröße, Objektalter oder einer Kombination dieser Filter basiert.

Sie können S3-Storage-Lens-Gruppen verwenden, um detaillierte Einblicke in große freigegebene Buckets wie Data Lakes zu erhalten und fundiertere Geschäftsentscheidungen zu treffen. Sie können beispielsweise die Speicherzuweisung rationalisieren und die Erstellung von Kostenberichten optimieren, indem Sie die Speichernutzung für einzelne Projekte und Kostenstellen innerhalb eines oder mehrerer Buckets nach bestimmten Objektgruppen aufschlüsseln.

Zur Nutzung von S3-Storage-Lens-Gruppen müssen Sie das Dashboard aktualisieren, um erweiterte Metriken und Empfehlungen zu verwenden. Weitere Informationen zur Verwendung von S3-Storage-Lens-Gruppen finden Sie unter [the section called “Arbeiten mit S3-Storage-Lens-Gruppen”](#).

Aktualisieren eines Amazon S3-Storage-Lens-Dashboards

Verwenden Sie die folgenden Schritte, um ein Amazon S3-Storage-Lens-Dashboard auf der Amazon S3-Konsole zu aktualisieren.

Schritt 1: Aktualisieren des Dashboard-Umfangs

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens, Dashboards aus.
3. Wählen Sie das Dashboard aus, das Sie bearbeiten möchten, und wählen Sie dann Bearbeiten.

Die Seite Edit dashboard (Dashboard bearbeiten) wird geöffnet.


Note

Folgendes können Sie nicht ändern:

- Den Dashboard-Namen
- Die Heimatregion
- Der Bereich des Standard-Dashboards, das für den Speicher Ihres gesamten Kontos eingerichtet ist


4. (Optional) Auf der Dashboard-Konfigurationsseite im Abschnitt General (Allgemein) können Sie Ihr Dashboard aktualisieren und Markierungen hinzufügen.

Sie können Markierungen verwenden, um Berechtigungen für Ihr Dashboard zu verwalten und die Kosten für S3 Storage Lens nachzuverfolgen. Weitere Informationen finden Sie unter [Steuern des Zugriffs mit Ressourcentags](#) im IAM-Benutzerhandbuch und [unter AWS-generierte Kostenzuordnungstags](#) im AWS Billing -Benutzerhandbuch.

 Note

Sie können Ihrer Dashboard-Konfiguration bis zu 50 Markierungen hinzufügen.

5. Führen Sie im Abschnitt Dashboard-Bereich die folgenden Schritte aus:
 - a. Aktualisieren Sie die Regionen und Buckets, die S3 Storage Lens im Dashboard ein- oder ausschließen soll.

 Note

- Sie können Regionen und Buckets entweder ein- oder ausschließen. Diese Option ist nur auf Regionen beschränkt, wenn Sie Dashboards auf Organisationsebene für alle Mitgliedskonten in Ihrer Organisation erstellen.
- Sie können bis zu 50 Buckets auswählen, die Sie einschließen oder ausschließen möchten.

- b. Aktualisieren Sie die Buckets in den ausgewählten Regionen, die S3 Storage Lens ein- oder ausschließen soll. Sie können Buckets entweder ein- oder ausschließen. Beides ist nicht möglich. Diese Option ist nicht vorhanden, wenn Sie Dashboards auf Organisationsebene erstellen.

Schritt 2: Aktualisieren der Metrikauswahl


1. Wählen Sie im Abschnitt Metrikauswahl den Metriktyp aus, den Sie für dieses Dashboard aggregieren möchten.
 - Wählen Sie Free metrics (Kostenlose Metriken) aus, um kostenlose Metriken einzuschließen, die auf Bucket-Ebene aggregiert und 14 Tage lang für Abfragen verfügbar sind.

- Wählen Sie **Advanced metrics and recommendations** (Erweiterte Metriken und Empfehlungen) aus, um erweiterte Metriken und andere erweiterte Optionen zu aktivieren. Zu diesen Optionen gehören erweiterte Präfixaggregation, CloudWatch Amazon-Veröffentlichung und kontextuelle Empfehlungen. Daten stehen für Abfragen für 15 Monate zur Verfügung. Erweiterte Metriken und Empfehlungen sind mit zusätzlichen Kosten verbunden. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Weitere Informationen zu erweiterten und zu kostenlosen Metriken finden Sie unter [Metrikauswahl](#).

2. Wählen Sie unter **Advanced metrics and recommendations features** (Erweiterte Metrik- und Empfehlungsfunktionen) die Optionen aus, Sie aktivieren möchten:

- **Advanced metrics** (Erweiterte Metriken)
- **CloudWatch Veröffentlichen**
- **Präfixzusammenfassung**

 **Important**

Wenn Sie die Präfixaggregation für Ihre S3 Storage Lens-Konfiguration aktivieren, werden Metriken auf Präfixebene nicht veröffentlicht. CloudWatch Nur S3 Storage Lens-Metriken auf Bucket-, Konto- und Organisationsebene werden veröffentlicht. CloudWatch

3. Wenn Sie **Advanced Metrics** (Erweiterte Metriken) aktiviert haben, wählen Sie die **Advanced metrics categories** (Erweiterte Metrikkategorien) aus, die Sie in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten:

- **Metriken für Aktivitäten**
- **Detailed status code metrics** (Detaillierte Statuscode-Metriken)
- **Advanced cost optimization metrics** (Erweiterte Kostenoptimierungsmetriken)
- **Advanced data protection metrics** (Erweiterte Datensicherheitsmetriken)

Weitere Informationen zu Metrikkategorien finden Sie unter [Metrikkategorien](#). Eine vollständige Liste der Metriken finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

4. Wenn Sie die Präfixaggregation aktivieren möchten, konfigurieren Sie Folgendes:

- a. Wählen Sie die Mindestgröße für den Präfixschwellenwert für dieses Dashboard aus.

Ein Präfixschwellenwert von 5 Prozent gibt beispielsweise an, dass Präfixe, die einen Anteil des gesamten Bucket-Speichers von 5 Prozent oder mehr ausmachen, aggregiert werden.

- b. Wählen Sie die Präfixtiefe aus.

Diese Einstellung gibt die maximale Anzahl von Ebenen an, bis zu denen die Präfixe ausgewertet werden. Die Präfixtiefe muss weniger als 10 betragen.

- c. Geben Sie ein Präfixtrennzeichen ein.

Dies ist der Wert, mit dem die einzelnen Präfixebenen identifiziert werden. Der Standardwert in Amazon S3 ist das Zeichen /, in Ihrer Speicherstruktur können jedoch ggf. andere Trennzeichen verwendet werden.

(Optional) Schritt 3: Exportieren von Metriken für das Dashboard

1. Wählen Sie im Abschnitt Metrics export (Metrik-Export) die Option Enable (Aktivieren) aus, um einen Metrik-Export zu erstellen, der jeden Tag in einem Ziel-Bucket Ihrer Wahl abgelegt wird. Wenn Sie den Metrik-Export deaktivieren möchten, wählen Sie Disable (Deaktivieren) aus.

Der Metrik-Export liegt im CSV- oder Apache Parquet-Format vor. Er enthält den gleichen Datenumfang wie Ihr S3-Storage-Lens-Dashboard ohne die Empfehlungen.

2. Wenn diese Option aktiviert ist, wählen Sie das Ausgabeformat für den täglichen Metrik-Export aus: CSV oder Apache Parquet.

Parquet ist ein Open-Source-Dateiformat für Hadoop, das verschachtelte Daten in einem flachen Spaltenformat speichert.

3. Wählen Sie den S3-Ziel-Bucket für den Export Ihrer Metriken aus.

Sie können einen Bucket im aktuellen Konto des S3-Storage-Lens-Dashboards auswählen. Sie können auch einen anderen auswählen, AWS-Konto wenn Sie über die Berechtigungen für den Ziel-Bucket und die Konto-ID des Besitzers des Ziel-Buckets verfügen.

4. Wählen Sie den S3-Ziel-Bucket (Format: `s3://bucket-name/prefix`) aus.


Der Bucket muss sich in der Heimatregion Ihres S3-Storage-Lens-Dashboards befinden. In der S3-Konsole wird das Feld Destination bucket permission (Ziel-Bucket-Berechtigung) mit der Berechtigung angezeigt, die der Ziel-Bucket-Richtlinie von Amazon S3 hinzugefügt wird. Amazon S3 aktualisiert die Bucket-Richtlinie für den Ziel-Bucket, damit S3 Daten darin ablegen kann.

5. (Optional) Wenn Sie die serverseitige Verschlüsselung für Ihren Metriken aktivieren möchten, wählen Sie [Specify an encryption key](#) (Einen Verschlüsselungsschlüssel angeben) aus. Wählen Sie anschließend den Verschlüsselungstyp aus: Von Amazon S3 verwaltete Schlüssel (SSE-S3) oder AWS Key Management Service -Schlüssel (SSE-KMS).

Sie können entweder einen [von Amazon S3 verwalteten Schlüssel](#) (SSE-S3) und einen [AWS Key Management Service \(AWS KMS\)](#)-Schlüssel (SSE-KMS) auswählen.

6. (Optional) Um einen AWS KMS Schlüssel anzugeben, müssen Sie einen KMS-Schlüssel wählen oder einen Amazon Resource Name (ARN) -Schlüssel eingeben. Geben Sie unter AWS KMS -Schlüssel Ihren KMS-Schlüssel auf eine der folgenden Arten an:
 - Wenn Sie aus einer Liste verfügbarer KMS-Schlüssel auswählen möchten, wählen Sie Aus Ihren AWS KMS keys wählen und anschließend den KMS-Schlüssel in der Liste der verfügbaren Schlüssel aus.

Sowohl der Von AWS verwalteter Schlüssel (aws/s3) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter [Kundenschlüssel und AWS -Schlüssel](#) im Entwicklerhandbuch zu AWS Key Management Service .

 Note

Von AWS verwalteter Schlüssel (aws/S3) wird für die SSE-KMS-Verschlüsselung mit S3 Storage Lens nicht unterstützt.

- Wählen Sie zum Eingeben des KMS-Schlüssel-ARN [AWS KMS key -ARN eingeben](#) aus und geben Sie Ihren KMS-Schlüssel-ARN in das angezeigte Feld ein.
- Um einen neuen vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie [Create a KMS key](#) aus.

Wenn Sie einen kundenverwalteten Schlüssel auswählen, müssen Sie S3 Storage Lens in der AWS KMS -Schlüsselrichtlinie die Berechtigung zum Verschlüsseln erteilen. Weitere Informationen finden Sie unter [Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln](#).

Weitere Informationen zum Erstellen eines finden Sie AWS KMS key unter [Creating Keys](#) im AWS Key Management Service Developer Guide.

7. Wählen Sie [Änderungen speichern](#) aus.

Um sich einen besseren Überblick über Ihren Speicher zu verschaffen, können Sie eine oder mehrere S3-Storage-Lens-Gruppen erstellen und diese an das Dashboard anhängen. Eine Storage-Lens-Gruppe ist ein benutzerdefinierter Filter für Objekte, der auf Präfixen, Suffixen, Objekt-Tags, Objektgröße, Objektalter oder einer Kombination dieser Filter basiert.

Sie können S3-Storage-Lens-Gruppen verwenden, um detaillierte Einblicke in große freigegebene Buckets wie Data Lakes zu erhalten und fundiertere Geschäftsentscheidungen zu treffen. Sie können beispielsweise die Speicherzuweisung rationalisieren und die Erstellung von Kostenberichten optimieren, indem Sie die Speichernutzung für einzelne Projekte und Kostenstellen innerhalb eines oder mehrerer Buckets nach bestimmten Objektgruppen aufschlüsseln.

Zur Nutzung von S3-Storage-Lens-Gruppen müssen Sie das Dashboard aktualisieren, um erweiterte Metriken und Empfehlungen zu verwenden. Weitere Informationen zur Verwendung von S3-Storage-Lens-Gruppen finden Sie unter [the section called “Arbeiten mit S3-Storage-Lens-Gruppen”](#).

Deaktivieren oder Löschen von Amazon S3-Storage-Lens-Dashboards

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden.

Das Standard-Dashboard von Amazon S3 Storage Lens ist default-account-dashboard. Dieses Dashboard ist von Amazon S3 vorkonfiguriert, damit Sie zusammengefasste Erkenntnisse und Trends der aggregierten kostenlosen und erweiterten Metriken Ihres gesamten Kontos in der Konsole visualisieren können. Sie können den Konfigurationsbereich des Standard-Dashboards nicht ändern, können aber die Metrikauswahl von den kostenlosen Metriken auf die kostenpflichtigen erweiterten Metriken und Empfehlungen upgraden, den optionalen Metrikexport konfigurieren oder sogar das Standard-Dashboard deaktivieren. Das Standard-Dashboard kann nicht gelöscht werden.

Sie können ein Amazon S3-Storage-Lens-Dashboard in der Amazon S3-Konsole löschen oder deaktivieren. Durch das Deaktivieren oder Löschen eines Dashboards wird verhindert, dass es weitere Metriken generiert. Ein deaktiviertes Dashboard behält seine Konfigurationsinformationen, sodass es bei erneuter Aktivierung einfach wieder genutzt werden kann. Ein deaktiviertes Dashboard behält seine Verlaufsdaten bei, bis es nicht mehr für Abfragen verfügbar ist.

Die Daten für die Auswahl der kostenlosen Metriken stehen 14 Tage lang für Abfragen zur Verfügung. Die Daten für die Auswahl der erweiterten Metriken und Empfehlungen stehen 15 Monate für Abfragen zur Verfügung.

Deaktivieren eines Amazon S3-Storage-Lens-Dashboards

So deaktivieren Sie ein S3-Storage-Lens-Dashboard

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie deaktivieren möchten, und gehen Sie dann oben in der Liste auf Deaktivieren.
4. Bestätigen Sie auf der Bestätigungsseite, dass Sie das Dashboard deaktivieren möchten, indem Sie den Namen des Dashboards in das Textfeld eingeben und dann Bestätigen auswählen.

Löschen eines Amazon S3-Storage-Lens-Dashboards

Note

Das Standard-Dashboard können Sie nicht löschen. Eine Deaktivierung ist jedoch möglich. Bevor Sie ein selbst erstelltes Dashboard löschen, sollten Sie Folgendes beachten:

- Als Alternative zum Löschen können Sie das Dashboard deaktivieren, damit es in Zukunft wieder aktiviert werden kann. Weitere Informationen finden Sie unter [Deaktivieren eines Amazon S3-Storage-Lens-Dashboards](#).
- Durch das Löschen eines Dashboards werden alle damit verbundenen Konfigurationseinstellungen gelöscht.
- Sobald Sie ein Dashboard löschen, können die historischen Metrikdaten nicht mehr abgerufen werden. Diese historischen Daten werden weiterhin für eine Dauer von 15 Monaten aufbewahrt. Wenn Sie wieder auf diese Daten zugreifen möchten, müssen Sie ein neues Dashboard mit dem gleichen Namen erstellen, das sich in derselben Heimatregion befindet wie das gelöschte Dashboard.

So löschen Sie ein S3-Storage-Lens-Dashboard

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie löschen möchten, und gehen Sie dann oben in der Liste auf Löschen.
4. Bestätigen Sie auf der Seite Dashboards löschen, dass Sie das Dashboard löschen möchten, indem Sie den Namen des Dashboards in das Textfeld eingeben. Wählen Sie dann Confirm (Bestätigen) aus.

Arbeiten mit AWS Organizations , um Dashboards auf Organisationsebene zu erstellen

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden.

Das Standard-Dashboard von Amazon S3 Storage Lens ist default-account-dashboard. Dieses Dashboard ist von Amazon S3 vorkonfiguriert, damit Sie zusammengefasste Erkenntnisse und Trends der aggregierten kostenlosen und erweiterten Metriken Ihres gesamten Kontos in der Konsole visualisieren können. Sie können den Konfigurationsbereich des Standard-Dashboards nicht ändern, können aber die Metrikauswahl von den kostenlosen Metriken auf die kostenpflichtigen erweiterten Metriken und Empfehlungen upgraden, den optionalen Metrikexport konfigurieren oder sogar das Standard-Dashboard deaktivieren. Das Standard-Dashboard kann nicht gelöscht werden.

Sie können auch zusätzliche S3 Storage Lens-Dashboards erstellen, die sich auf bestimmte AWS-Regionen S3-Bereiche oder andere Bereiche AWS-Konten in Ihrem Unternehmen konzentrieren.

Ein S3-Storage-Lens-Dashboard bietet umfassende Informationen zum zugehörigen Speicherbereich. In einem Dashboard werden über 30 Metriken visualisiert, die Trends und andere Informationen darstellen, einschließlich Speicherzusammenfassung, Kosteneffizienz, Datenschutz und Aktivitäten.

Amazon S3 Storage Lens kann verwendet werden, um Speichermetriken und Nutzungsdaten für alle Konten zu sammeln, die Teil Ihrer AWS Organizations Hierarchie sind. Dazu müssen Sie den vertrauenswürdigen Zugriff von AWS Organizations S3 Storage Lens mithilfe Ihres AWS Organizations Verwaltungskontos verwenden und aktivieren.

Wenn der vertrauenswürdige Zugriff aktiviert ist, können Sie anderen Konten in Ihrer Organisation den delegierten Administratorzugriff gewähren. Diese Konten können dann organisationsweite Dashboards und Konfigurationen für S3 Storage Lens erstellen. Weitere Informationen zum Aktivieren des vertrauenswürdigen Zugriffs finden Sie in [Amazon S3 Lens und AWS Organizations](#) im AWS Organizations -Benutzerhandbuch.

Die folgenden Konsolensteuerungen sind nur für die AWS Organizations Verwaltungskonten verfügbar.

Aktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens in Ihrer Organisation

Durch die Aktivierung des vertrauenswürdigen Zugriffs kann Amazon S3 Storage Lens über AWS Organizations API-Operationen auf Ihre AWS Organizations Hierarchie, Mitgliedschaft und Struktur zugreifen. S3 Storage Lens wird zu einem vertrauenswürdigen Service für die Struktur Ihrer gesamten Organisation. Immer wenn eine Dashboard-Konfiguration erstellt wird, kann der Service in den Verwaltungskonten oder den delegierten Administratorkonten Ihrer Organisation serviceverknüpfte Rollen erstellen.

Die serviceverknüpfte Rolle gewährt S3 Storage Lens Berechtigungen, um Organisationen zu beschreiben, Konten aufzulisten, eine Liste des -Servicezugriffs für die Organisationen zu verifizieren und die delegierten Administratoren der Organisation abzurufen. Auf diese Weise kann S3 Storage Lens kontoübergreifende Speichernutzungs- und Aktivitätsmetriken für Dashboards in Konten Ihrer Organisation erfassen.

Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens](#).

Note

- Vertrauenswürdiger Zugriff kann nur im Verwaltungskonto aktiviert werden.
- S3-Storage-Lens-Dashboards oder -Konfigurationen für Ihre Organisation können nur über das Verwaltungskonto und von delegierten Administratoren erstellt werden.

So aktivieren Sie den vertrauenswürdigen Zugriff für S3 Storage Lens

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und Organization settings (Einstellungen für Organisation) aus.
3. Wählen Sie unter Organizations-Zugriff die Option Bearbeiten aus.

Die Seite Organizations-Zugriff wird geöffnet. Hier können Sie den vertrauenswürdigen Zugriff für S3 Storage Lens aktivieren. Dadurch können Sie und alle anderen Kontoinhaber, die Sie als delegierte Administratoren hinzufügen, Dashboards für alle Konten und Speicher in Ihrer Organisation erstellen.

Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens in Ihrer Organisation

Durch die Deaktivierung des vertrauenswürdigen Zugriffs wird die Funktionalität von S3 Storage Lens auf die Kontoebene beschränkt. Jeder Kontoinhaber sieht nur die S3-Storage-Lens-Vorteile, die auf den Bereich seines Kontos beschränkt sind. Die Funktionen für die gesamte Organisation werden nicht angezeigt. Alle Dashboards, die einen vertrauenswürdigen Zugriff erfordern, werden nicht mehr aktualisiert. Die Dashboards können ihre historischen Daten jedoch gemäß dem jeweils geltenden [Zeitraum, in dem Daten für Abfragen verfügbar sind](#) abfragen.

Wenn der delegierte Administratorzugriff eines Kontos entfernt wird, kann der Kontoinhaber nur noch auf Kontoebene auf die Dashboard-Metriken von S3 Storage Lens zugreifen. Alle von ihnen erstellten Organisations-Dashboards werden nicht mehr aktualisiert, aber sie können ihre historischen Daten für den Zeitraum abfragen, [in dem sie für Abfragen verfügbar sind](#).

Note

- Durch die Deaktivierung des vertrauenswürdigen Zugriffs werden auch automatisch alle Dashboards auf Organisationsebene deaktiviert, da S3 Storage Lens keinen vertrauenswürdigen Zugriff mehr auf die Organisationskonten hat, um Speichermetriken zu erfassen und zu aggregieren.
- Die Verwaltungs- und Stellvertreter-Administratorkonten können weiterhin die historischen Daten für diese deaktivierten Dashboards sehen und diese Daten abfragen, solange sie verfügbar sind.

So deaktivieren Sie den vertrauenswürdigen Zugriff für S3 Storage Lens

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und Organization settings (Einstellungen für Organisation) aus.
3. Wählen Sie unter Organizations-Zugriff die Option Bearbeiten aus.

Die Seite Organizations-Zugriff wird geöffnet. Hier können Sie den vertrauenswürdigen Zugriff für S3 Storage Lens deaktivieren.

Registrieren von delegierten Administratoren für S3 Storage Lens

Nachdem Sie den vertrauenswürdigen Zugriff aktiviert haben, können Sie andere Konten in Ihrer Organisation für den delegierten Administratorzugriff registrieren. Wenn ein Konto als delegierter Administrator registriert ist, erhält das Konto die Autorisierung für den Zugriff auf alle schreibgeschützten API-Operationen AWS Organizations . Das macht die Mitglieder und Strukturen Ihrer Organisation sichtbar, sodass die delegierten Administratoren in Ihrem Namen S3-Storage-Lens-Dashboards erstellen können.

So registrieren Sie delegierte Administratoren für S3 Storage Lens

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und Organization settings (Einstellungen für Organisation) aus.
3. Wählen Sie im Abschnitt Delegierter Zugriff für Konten die Option Konto hinzufügen aus.

Die Seite Delegierter Administratorzugriff wird geöffnet. Hier können Sie eine AWS-Konto -ID als delegierten Administrator hinzufügen, um Dashboards auf Organisationsebene für alle Konten und Speicher in Ihrer Organisation zu erstellen.

Aufheben der Registrierung delegierter Administratoren für S3 Storage Lens

Sie können den delegierten Administratorzugriff für Konten in Ihrer Organisation wieder entfernen. Wenn ein Konto als delegierter Administrator abgemeldet wird, verliert das Konto die Autorisierung für den Zugriff auf alle schreibgeschützten AWS Organizations API-Operationen, die den Mitgliedern und Strukturen Ihrer Organisation Einblick gewähren.

Note

- Durch das Aufheben der Registrierung eines delegierten Administrators werden automatisch auch alle Dashboards auf Organisationsebene deaktiviert, die der delegierte Administrator erstellt hat.
- Die delegierten Administratorkonten können weiterhin die historischen Daten für diese deaktivierten Dashboards gemäß dem jeweiligen Zeitraum anzeigen, in dem die Daten für Abfragen verfügbar sind.

So heben Sie die Registrierung von Konten für den delegierten Administratorzugriff auf

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Storage Lens und Organization settings (Einstellungen für Organisation) aus.
3. Wählen Sie im Abschnitt Konten mit delegiertem Zugriff die Konto-ID aus, deren Registrierung Sie aufheben möchten, und gehen Sie dann auf Entfernen.

Amazon S3-Storage-Lens-Beispiele für die Verwendung der AWS CLI.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden. Weitere Informationen finden Sie unter [Bewertung der Speicheraktivität und -nutzung mit Amazon S3 Storage Lens](#).

In den folgenden Beispielen wird gezeigt, wie Sie S3 Storage Lens mit der AWS Command Line Interface verwenden.

Themen

- [Hilfsdateien für die Verwendung von Amazon S3 Storage Lens](#)

- [Verwenden von Amazon S3-Storage-Lens-Konfigurationen mit der AWS CLI](#)
- [Beispiele für das Verwenden von Amazon S3 Storage Lens mit AWS Organizations unter Verwendung der AWS CLI](#)

Hilfsdateien für die Verwendung von Amazon S3 Storage Lens

Verwenden Sie die folgenden JSON-Dateien und die zugehörigen wichtigen Eingaben für Ihre Beispiele.

Beispiel für S3-Storage-Lens-Konfiguration in JSON

Example **config.json**

Die Datei `config.json` enthält Details einer S3-Storage-Lens-Konfiguration auf Organisationsebene mit erweiterten Metriken und Empfehlungen. Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Note

Für erweiterte Metriken und Empfehlungen fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter [Erweiterte Metriken und Empfehlungen](#).

```
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
  Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled": true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled": true
    },
    "DetailedStatusCodesMetrics": {
```



```

    "IsEnabled":true
  },
  "BucketLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
    "PrefixLevel":{
      "StorageMetrics":{
        "IsEnabled":true,
        "SelectionCriteria":{
          "MaxDepth":5,
          "MinStorageBytesPercentage":1.25,
          "Delimiter":"/"
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
      "eu-west-1"
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
configurations.
      "arn:aws:s3:::source_bucket1"
    ]
  },
  "IsEnabled": true, //Whether the configuration is enabled
  "DataExport": { //Details about the metrics export
    "S3BucketDestination": {
      "OutputSchemaVersion": "V_1",
      "Format": "CSV", //You can add "Parquet" if you prefer.
      "AccountId": "111122223333",
      "Arn": "arn:aws:s3:::destination-bucket-name", // The destination bucket for your
metrics export must be in the same Region as your S3 Storage Lens configuration.

```

```
    "Prefix": "prefix-for-your-export-destination",
    "Encryption": {
      "SSE3": {}
    },
    "CloudWatchMetrics": {
      "IsEnabled": true
    }
  }
}
```

Beispiel für S3-Storage-Lens-Konfiguration mit Storage-Lens-Gruppen in JSON

Example **config.json**

Die Datei `config.json` enthält die Details, die Sie bei Verwendung von Storage-Lens-Gruppen auf die Storage-Lens-Konfiguration anwenden möchten. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

Um alle Storage-Lens-Gruppen an das Dashboard anzuhängen, aktualisieren Sie die Storage-Lens-Konfiguration mit der folgenden Syntax:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true
      },
      "StorageLensGroupLevel": {},
      "IsEnabled": true
    }
  }
}
```

Verwenden Sie die folgende Syntax, wenn Sie nur zwei Storage-Lens-Gruppen (*slg-1* und *slg-2*) in die Storage-Lens-Dashboard-Konfiguration aufnehmen möchten:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled": true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled": true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled": true
      },
      "StorageLensGroupLevel": {
        "SelectionCriteria": {
          "Include": [
            "arn:aws:s3:us-east-1:111122223333:storage-lens-group/sl-1",
            "arn:aws:s3:us-east-1:444455556666:storage-lens-group/sl-2"
          ]
        }
      },
      "IsEnabled": true
    }
  }
}
```

Verwenden Sie die folgende Syntax, um bestimmte Storage-Lens-Gruppen vom Anhängen an die Dashboard-Konfiguration auszuschließen:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled": true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled": true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled": true
    }
  }
}
```

```

    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled": true
      },
    },
    "StorageLensGroupLevel": {
      "SelectionCriteria": {
        "Exclude": [
          "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
          "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
        ]
      },
    },
    "IsEnabled": true
  }
}

```

Beispiel für S3-Storage-Lens-Konfiguration mit Tags in JSON

Example **tags.json**

Die Datei `tags.json` enthält die Tags, die Sie auf Ihre S3-Storage-Lens-Konfiguration anwenden möchten. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```

[
  {
    "Key": "key1",
    "Value": "value1"
  },
  {
    "Key": "key2",
    "Value": "value2"
  }
]

```

Beispiel für S3-Storage-Lens-Konfiguration für IAM Berechtigungen

Example **permissions.json** – Spezifischer Dashboard-Name

Diese Beispielrichtlinie zeigt eine IAM-Datei `permissions.json` von S3 Storage Lens mit einem bestimmten Dashboard-Namen. Ersetzen Sie *value1*, *us-east-1*, *your-dashboard-name* und *example-account-id* durch Ihre eigenen Werte.

```

{

```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetStorageLensConfiguration",
      "s3>DeleteStorageLensConfiguration",
      "s3:PutStorageLensConfiguration"
    ],
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/key1": "value1"
      }
    },
    "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/your-
dashboard-name"
  }
]
}

```

Example **permissions.json** – Kein spezifischer Dashboard-Name

Diese Beispielrichtlinie zeigt eine IAM-Datei `permissions.json` von S3 Storage Lens ohne einen bestimmten Dashboard-Namen. Ersetzen Sie `value1`, `us-east-1` und `example-account-id` durch Ihre eigenen Werte.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetStorageLensConfiguration",
        "s3>DeleteStorageLensConfiguration",
        "s3:PutStorageLensConfiguration"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/key1": "value1"
        }
      },
      "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/*"
    }
  ]
}

```

```
    }  
  ]  
}
```

Verwenden von Amazon S3-Storage-Lens-Konfigurationen mit der AWS CLI

Sie können die AWS CLI verwenden, um S3-Storage-Lens-Konfigurationen aufzulisten, zu erstellen, zu löschen, abzurufen, zu markieren und zu aktualisieren. In den folgenden Beispielen werden die JSON-Hilfsdateien für wichtige Eingaben verwendet. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

Erstellen einer S3-Storage-Lens-Konfiguration

Example Erstellen einer S3-Storage-Lens-Konfiguration

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --  
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-  
configuration=file:///config.json --tags=file:///tags.json
```

Erstellen einer S3-Storage-Lens-Konfiguration ohne Markierungen

Example Erstellen einer S3-Storage-Lens-Konfiguration ohne Markierungen

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-  
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file:////  
config.json
```

Abrufen einer S3-Storage-Lens-Konfiguration

Example Abrufen einer S3-Storage-Lens-Konfiguration

```
aws s3control get-storage-lens-configuration --account-id=222222222222 --config-  
id=your-configuration-id --region=us-east-1
```

Auflisten von S3-Storage-Lens-Konfigurationen ohne das nächste Token

Example Auflisten von S3-Storage-Lens-Konfigurationen ohne das nächste Token

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-  
east-1
```

Listet S3-Storage-Lens-Konfigurationen auf

Example Listet S3-Storage-Lens-Konfigurationen auf

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-east-1 --next-token=abcdefghijkl234
```

Löschen einer S3-Storage-Lens-Konfiguration

Example Löschen einer S3-Storage-Lens-Konfiguration

```
aws s3control delete-storage-lens-configuration --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Hinzufügen von Markierungen zu einer S3-Storage-Lens-Konfiguration

Example Hinzufügen von Markierungen zu einer S3-Storage-Lens-Konfiguration

```
aws s3control put-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id --tags=file:///./tags.json
```

Abrufen von Markierungen für eine S3-Storage-Lens-Konfiguration

Example Abrufen von Markierungen für eine S3-Storage-Lens-Konfiguration

```
aws s3control get-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Löschen von Markierungen für eine S3-Storage-Lens-Konfiguration

Example Löschen von Markierungen für eine S3-Storage-Lens-Konfiguration

```
aws s3control delete-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Beispiele für das Verwenden von Amazon S3 Storage Lens mit AWS Organizations unter Verwendung der AWS CLI

Verwenden Sie Amazon S3 Storage Lens, um Speichermetriken und Nutzungsdaten für alle Konten zu erfassen, die Teil Ihrer AWS Organizations-Hierarchie sind. Weitere Informationen finden Sie unter

[Using Amazon S3 Storage Lens with AWS Organizations \(Verwenden von Amazon S3 Storage Lens mit AO\).](#)

Aktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

Example Aktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

```
aws organizations enable-aws-service-access --service-principal storage-lens.s3.amazonaws.com
```

Deaktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

Example Deaktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

```
aws organizations disable-aws-service-access --service-principal storage-lens.s3.amazonaws.com
```

Registrieren delegierter Organizations-Administratoren für S3 Storage Lens

Example Registrieren delegierter Organizations-Administratoren für S3 Storage Lens

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie **111122223333** durch die entsprechende AWS-Konto-ID.

```
aws organizations register-delegated-administrator --service-principal storage-lens.s3.amazonaws.com --account-id 111122223333
```

Registrierung delegierter Organizations-Administratoren für S3 Storage Lens aufheben

Example Registrierung delegierter Organizations-Administratoren für S3 Storage Lens aufheben

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie **111122223333** durch die entsprechende AWS-Konto-ID.

```
aws organizations deregister-delegated-administrator --service-principal storage-lens.s3.amazonaws.com --account-id 111122223333
```

Amazon-S3-Storage-Lens-Beispiele für die Verwendung des SDK for Java

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet

außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Storage-Lens-Gruppenebene zu generieren. Sie können zudem täglich einen Metrikexport im CSV- oder Parquet-Format an einen S3-Bucket senden. Weitere Informationen finden Sie unter [Bewertung der Speicheraktivität und -nutzung mit Amazon S3 Storage Lens](#).

In den folgenden Beispielen wird gezeigt, wie Sie S3 Storage Lens mit AWS SDK for Java verwenden.

Themen

- [Verwenden von Amazon S3-Storage-Lens-Konfigurationen mit dem SDK for Java](#)

Verwenden von Amazon S3-Storage-Lens-Konfigurationen mit dem SDK for Java

Sie können das SDK for Java verwenden, um S3-Storage-Lens-Konfigurationen aufzulisten, zu erstellen, abzurufen und zu aktualisieren. In den folgenden Beispielen werden die JSON-Hilfsdateien für wichtige Eingaben verwendet.

Themen

- [Erstellen und Aktualisieren einer S3-Storage-Lens-Konfiguration](#)
- [Löschen einer S3-Storage-Lens-Konfiguration](#)
- [Abrufen einer S3-Storage-Lens-Konfiguration](#)
- [Listet S3-Storage-Lens-Konfigurationen auf](#)
- [Hinzufügen von Markierungen zu einer S3-Storage-Lens-Konfiguration](#)
- [Abrufen von Markierungen für eine S3-Storage-Lens-Konfiguration](#)
- [Löschen von Markierungen für eine S3-Storage-Lens-Konfiguration](#)
- [Aktualisieren der Standardkonfiguration von S3 Storage Lens mit erweiterten Metriken und Empfehlungen](#)
- [Anhängen einer Storage-Lens-Gruppe an ein S3-Storage-Lens-Dashboard](#)
- [Verwenden von Amazon S3 Storage Lens mit AWS Organizations-Beispielen unter Verwendung von SDK für Java](#)

Erstellen und Aktualisieren einer S3-Storage-Lens-Konfiguration

Example Erstellen und Aktualisieren einer S3-Storage-Lens-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination
        bucket for your metrics export must be in the same Region as your S3 Storage Lens
        configuration.
    }
}
```

```
String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-  
abcdefgh";  
Format exportFormat = Format.CSV;  
  
try {  
    SelectionCriteria selectionCriteria = new SelectionCriteria()  
        .withDelimiter("/")  
        .withMaxDepth(5)  
        .withMinStorageBytesPercentage(10.0);  
    PrefixLevelStorageMetrics prefixStorageMetrics = new  
PrefixLevelStorageMetrics()  
        .withIsEnabled(true)  
        .withSelectionCriteria(selectionCriteria);  
    BucketLevel bucketLevel = new BucketLevel()  
        .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))  
        .withAdvancedCostOptimizationMetrics(new  
AdvancedCostOptimizationMetrics().withIsEnabled(true))  
        .withAdvancedDataProtectionMetrics(new  
AdvancedDataProtectionMetrics().withIsEnabled(true))  
        .withDetailedStatusCodesMetrics(new  
DetailedStatusCodesMetrics().withIsEnabled(true))  
        .withPrefixLevel(new  
PrefixLevel().withStorageMetrics(prefixStorageMetrics));  
    AccountLevel accountLevel = new AccountLevel()  
        .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))  
        .withAdvancedCostOptimizationMetrics(new  
AdvancedCostOptimizationMetrics().withIsEnabled(true))  
        .withAdvancedDataProtectionMetrics(new  
AdvancedDataProtectionMetrics().withIsEnabled(true))  
        .withDetailedStatusCodesMetrics(new  
DetailedStatusCodesMetrics().withIsEnabled(true))  
        .withBucketLevel(bucketLevel);  
  
    Include include = new Include()  
        .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))  
        .withRegions(Arrays.asList("us-west-2"));  
  
    StorageLensDataExportEncryption exportEncryption = new  
StorageLensDataExportEncryption()  
        .withSSES3(new SSES3());  
    S3BucketDestination s3BucketDestination = new S3BucketDestination()  
        .withAccountId(exportAccountId)  
        .withArn(exportBucketArn)  
        .withEncryption(exportEncryption)
```

```
        .withFormat(exportFormat)
        .withOutputSchemaVersion(OutputSchemaVersion.V_1)
        .withPrefix("Prefix");
CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
    .withIsEnabled(true);
StorageLensDataExport dataExport = new StorageLensDataExport()
    .withCloudWatchMetrics(cloudWatchMetrics)
    .withS3BucketDestination(s3BucketDestination);

StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
    .withArn(awsOrgARN);

StorageLensConfiguration configuration = new StorageLensConfiguration()
    .withId(configurationId)
    .withAccountLevel(accountLevel)
    .withInclude(include)
    .withDataExport(dataExport)
    .withAwsOrg(awsOrg)
    .withIsEnabled(true);

List<StorageLensTag> tags = Arrays.asList(
    new StorageLensTag().withKey("key-1").withValue("value-1"),
    new StorageLensTag().withKey("key-2").withValue("value-2")
);

AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
    .withCredentials(new ProfileCredentialsProvider())
    .withRegion(US_WEST_2)
    .build();

s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
    .withAccountId(sourceAccountId)
    .withConfigId(configurationId)
    .withStorageLensConfiguration(configuration)
    .withTags(tags)
);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
```

```
        e.printStackTrace();
    }
}
}
```

Löschen einer S3-Storage-Lens-Konfiguration

Example Löschen einer S3-Storage-Lens-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.deleteStorageLensConfiguration(new
DeleteStorageLensConfigurationRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
        }
    }
}
```

```
        e.printStackTrace();
    }
}
}
```

Abrufen einer S3-Storage-Lens-Konfiguration

Example Abrufen einer S3-Storage-Lens-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationResult;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            final StorageLensConfiguration configuration =
                s3ControlClient.getStorageLensConfiguration(new
                GetStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                ).getStorageLensConfiguration();

            System.out.println(configuration.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
```

```
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

Listet S3-Storage-Lens-Konfigurationen auf

Example Listet S3-Storage-Lens-Konfigurationen auf

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationEntry;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationsRequest;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class ListDashboard {

    public static void main(String[] args) {
        String sourceAccountId = "Source Account ID";
        String nextToken = "nextToken";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            final List<ListStorageLensConfigurationEntry> configurations =
                s3ControlClient.listStorageLensConfigurations(new
ListStorageLensConfigurationsRequest()
                    .withAccountId(sourceAccountId)
```

```
        .withNextToken(nextToken)
        ).getStorageLensConfigurationList();

        System.out.println(configurations.toString());
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

Hinzufügen von Markierungen zu einer S3-Storage-Lens-Konfiguration

Example Hinzufügen von Markierungen zu einer S3-Storage-Lens-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
    com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";

        try {
            List<StorageLensTag> tags = Arrays.asList(
```



```
        new StorageLensTag().withKey("key-1").withValue("value-1"),
        new StorageLensTag().withKey("key-2").withValue("value-2")
    );

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.putStorageLensConfigurationTagging(new
PutStorageLensConfigurationTaggingRequest()
        .withAccountId(sourceAccountId)
        .withConfigId(configurationId)
        .withTags(tags)
    );
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

Abrufen von Markierungen für eine S3-Storage-Lens-Konfiguration

Example Abrufen von Markierungen für eine S3-Storage-Lens-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;
import
    com.amazonaws.services.s3control.model.GetStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.List;
```

```
import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            final List<StorageLensTag> s3Tags = s3ControlClient
                .getStorageLensConfigurationTagging(new
                GetStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                ).getTags();

            System.out.println(s3Tags.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Löschen von Markierungen für eine S3-Storage-Lens-Konfiguration

Example Löschen von Markierungen für eine S3-Storage-Lens-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
```

```
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
    com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationTaggingRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.deleteStorageLensConfigurationTagging(new
DeleteStorageLensConfigurationTaggingRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Aktualisieren der Standardkonfiguration von S3 Storage Lens mit erweiterten Metriken und Empfehlungen

Example Aktualisieren der Standardkonfiguration von S3 Storage Lens mit erweiterten Metriken und Empfehlungen

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateDefaultConfigWithPaidFeatures {

    public static void main(String[] args) {
        String configurationId = "default-account-dashboard"; // This configuration ID
        cannot be modified.
        String sourceAccountId = "Source Account ID";

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                .withDelimiter("/")
                .withMaxDepth(5)
                .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
            PrefixLevelStorageMetrics()
                .withIsEnabled(true)
                .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
```

```
        .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
        .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
    AccountLevel accountLevel = new AccountLevel()
        .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
        .withBucketLevel(bucketLevel);

    StorageLensConfiguration configuration = new StorageLensConfiguration()
        .withId(configurationId)
        .withAccountLevel(accountLevel)
        .withIsEnabled(true);

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
        .withAccountId(sourceAccountId)
        .withConfigId(configurationId)
        .withStorageLensConfiguration(configuration)
    );

} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Note

Für erweiterte Metriken und Empfehlungen fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter [Erweiterte Metriken und Empfehlungen](#).

Anhängen einer Storage-Lens-Gruppe an ein S3-Storage-Lens-Dashboard

Example Anhängen aller Storage-Lens-Gruppen an ein Dashboard

Im folgenden SDK-für-Java-Beispiel werden alle Storage-Lens-Gruppen im Konto *111122223333* an das Dashboard *DashBoardConfigurationId* angehängt.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWithStorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel();

            AccountLevel accountLevel = new AccountLevel()
                .withBucketLevel(new BucketLevel())
                .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                .withId(configurationId)
                .withAccountLevel(accountLevel)
                .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();
```

```
s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
    .withAccountId(sourceAccountId)
    .withConfigId(configurationId)
    .withStorageLensConfiguration(configuration)
);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Example Anhängen von zwei Storage-Lens-Gruppen an ein Dashboard

Im folgenden AWS SDK for Java-Beispiel werden zwei Storage-Lens-Gruppen (*StorageLensGroupName1* und *StorageLensGroupName2*) an das Dashboard *ExampleDashboardConfigurationId* angehängt.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroups {
    public static void main(String[] args) {
```

```
String configurationId = "ExampleDashboardConfigurationId";
String storageLensGroupName1 = "StorageLensGroupName1";
String storageLensGroupName2 = "StorageLensGroupName2";
String sourceAccountId = "111122223333";

try {
    StorageLensGroupLevelSelectionCriteria selectionCriteria = new
StorageLensGroupLevelSelectionCriteria()
        .withInclude(
            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName1,
            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName2);

    System.out.println(selectionCriteria);
    StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
        .withSelectionCriteria(selectionCriteria);

    AccountLevel accountLevel = new AccountLevel()
        .withBucketLevel(new BucketLevel())
        .withStorageLensGroupLevel(storageLensGroupLevel);

    StorageLensConfiguration configuration = new StorageLensConfiguration()
        .withId(configurationId)
        .withAccountLevel(accountLevel)
        .withIsEnabled(true);

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
        .withAccountId(sourceAccountId)
        .withConfigId(configurationId)
        .withStorageLensConfiguration(configuration)
    );
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
```



```
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

Example Anhängen aller Storage-Lens-Gruppen mit Ausschlüssen

Im folgenden SDK-für-Java-Beispiel werden bis auf zwei angegebene Storage-Lens-Gruppen (*StorageLensGroupName1* und *StorageLensGroupName2*) alle Storage-Lens-Gruppen an das Dashboard *ExampleDashboardConfigurationId* angehängt.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroupsExcluded {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
StorageLensGroupLevelSelectionCriteria()
                .withInclude(
                    "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName1,
```

```
        "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
+ ":storage-lens-group/" + storageLensGroupName2);

    System.out.println(selectionCriteria);
    StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
        .withSelectionCriteria(selectionCriteria);

    AccountLevel accountLevel = new AccountLevel()
        .withBucketLevel(new BucketLevel())
        .withStorageLensGroupLevel(storageLensGroupLevel);

    StorageLensConfiguration configuration = new StorageLensConfiguration()
        .withId(configurationId)
        .withAccountLevel(accountLevel)
        .withIsEnabled(true);

    AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(US_WEST_2)
        .build();

    s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
        .withAccountId(sourceAccountId)
        .withConfigId(configurationId)
        .withStorageLensConfiguration(configuration)
    );
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Verwenden von Amazon S3 Storage Lens mit AWS Organizations-Beispielen unter Verwendung von SDK für Java

Verwenden Sie Amazon S3 Storage Lens, um Speichermetriken und Nutzungsdaten für alle Konten zu erfassen, die Teil Ihrer AWS Organizations-Hierarchie sind. Weitere Informationen finden Sie unter [Using Amazon S3 Storage Lens with AWS Organizations \(Verwenden von Amazon S3 Storage Lens mit AO\)](#).

Themen

- [Aktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens](#)
- [Deaktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens](#)
- [Registrieren delegierter Organizations-Administratoren für S3 Storage Lens](#)
- [Registrierung delegierter Organizations-Administratoren für S3 Storage Lens aufheben](#)

Aktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

Example Aktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.EnableAWSServiceAccessRequest;

public class EnableOrganizationsTrustedAccess {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

    public static void main(String[] args) {
        try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.enableAWSServiceAccess(new
EnableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
```

```

    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but AWS Organizations couldn't
process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // AWS Organizations couldn't be contacted for a response, or the client
        // couldn't parse the response from AWS Organizations.
        e.printStackTrace();
    }
}
}
}

```

Deaktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

Example Deaktivieren des vertrauenswürdigen Zugriffs von Organizations für S3 Storage Lens

```

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.DisableAWSServiceAccessRequest;

public class DisableOrganizationsTrustedAccess {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

    public static void main(String[] args) {
        try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            // Make sure to remove any existing delegated administrator for S3 Storage
Lens
            // before disabling access; otherwise, the request will fail.
            organizationsClient.disableAWSServiceAccess(new
DisableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {

```

```
        // The call was transmitted successfully, but AWS Organizations couldn't
process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // AWS Organizations couldn't be contacted for a response, or the client
        // couldn't parse the response from AWS Organizations.
        e.printStackTrace();
    }
}
}
```

Registrieren delegierter Organizations-Administratoren für S3 Storage Lens

Example Registrieren delegierter Organizations-Administratoren für S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
    com.amazonaws.services.organizations.model.RegisterDelegatedAdministratorRequest;

public class RegisterOrganizationsDelegatedAdministrator {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

    public static void main(String[] args) {
        try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.registerDelegatedAdministrator(new
RegisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
```

```

        // The call was transmitted successfully, but AWS Organizations couldn't
process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // AWS Organizations couldn't be contacted for a response, or the client
        // couldn't parse the response from AWS Organizations.
        e.printStackTrace();
    }
}
}
}

```

Registrierung delegierter Organizations-Administratoren für S3 Storage Lens aufheben

Example Registrierung delegierter Organizations-Administratoren für S3 Storage Lens aufheben

```

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
    com.amazonaws.services.organizations.model.DeregisterDelegatedAdministratorRequest;

public class DeregisterOrganizationsDelegatedAdministrator {
    private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

    public static void main(String[] args) {
        try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.deregisterDelegatedAdministrator(new
DeregisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {

```

```
        // The call was transmitted successfully, but AWS Organizations couldn't
process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // AWS Organizations couldn't be contacted for a response, or the client
        // couldn't parse the response from AWS Organizations.
        e.printStackTrace();
    }
}
}
```

Arbeiten mit S3-Storage-Lens-Gruppen

Eine Amazon-S3-Storage-Lens-Gruppe aggregiert Metriken mithilfe benutzerdefinierter Filter auf der Grundlage von Objektmetadaten. Mit Storage-Lens-Gruppen können Sie Details zu den Eigenschaften Ihrer Daten anzeigen, etwa die Objektverteilung nach Alter, die gängigsten Dateitypen usw. Sie können Metriken beispielsweise nach Objekt-Tag filtern, um die am schnellsten wachsenden Datensätze zu ermitteln, oder den Speicher auf Grundlage der Objektgröße und des Objektalters visualisieren, um die Speicherarchivierungsstrategie festzulegen. Folglich vermitteln Amazon-S3-Storage-Lens-Gruppen Ihnen ein besseres Verständnis des S3-Speichers und unterstützen Sie bei seiner Optimierung.

Wenn Sie Storage-Lens-Gruppen verwenden, können Sie S3-Storage-Lens-Metriken anhand von Objektmetadaten wie Präfixen, Suffixen, [Objekt-Tags](#), Objektgröße oder Objektalter analysieren und filtern. Diese Filter lassen sich auch kombinieren. Nachdem Sie eine Storage-Lens-Gruppe an das S3-Storage-Lens-Dashboard angehängt haben, können Sie die nach Amazon-S3-Storage-Lens-Gruppen aggregierten S3-Storage-Lens-Metriken direkt im Dashboard anzeigen.

Filtern Sie Messwerte beispielsweise auch nach Objektgröße oder Altersklassen, um festzustellen, welcher Speicheranteil aus kleinen Objekten besteht. Sie können diese Informationen dann mit S3 Intelligent-Tiering oder im S3-Lebenszyklus verwenden, um kleine Objekte zur Kosten- und Speicheroptimierung in verschiedene Speicherklassen zu überführen.

Themen

- [Funktionsweise von S3-Storage-Lens-Gruppen](#)
- [Verwenden von Storage-Lens-Gruppen](#)

Funktionsweise von S3-Storage-Lens-Gruppen

Mit Storage-Lens-Gruppen lassen sich Metriken mithilfe benutzerdefinierter Filter auf der Grundlage von Objektmetadaten aggregieren. Wenn Sie einen benutzerdefinierten Filter definieren, können Sie Präfixe, Suffixe, Objekt-Tags, Objektgrößen, Objektalter oder eine Kombination dieser benutzerdefinierten Filter verwenden. Bei der Erstellung einer Storage-Lens-Gruppe können Sie zudem einen einzelnen Filter oder mehrere Filterbedingungen angeben. Zur Angabe mehrerer Filterbedingungen verwenden Sie den logischen Operator `And` oder `Or`.

Wenn Sie eine Storage-Lens-Gruppe erstellen und konfigurieren, dient die Storage-Lens-Gruppe selbst als benutzerdefinierter Filter im Dashboard, an das Sie die Gruppe anhängen. Im Dashboard können Sie dann den Storage-Lens-Gruppenfilter verwenden, um Speichermetriken auf Grundlage des benutzerdefinierten Filters abzurufen, den Sie in der Gruppe definiert haben.

Um die Daten für die Storage-Lens-Gruppe im S3-Storage-Lens-Dashboard anzuzeigen, müssen Sie die Gruppe nach der Erstellung an das Dashboard anhängen. Nachdem die Storage-Lens-Gruppe an das Storage-Lens-Dashboard angehängt wurde, erfasst das Dashboard innerhalb von 48 Stunden Metriken zur Speichernutzung. Sie können diese Daten dann im Storage-Lens-Dashboard visualisieren oder sie über einen Metrikexport exportieren. Falls Sie vergessen, eine Storage-Lens-Gruppe an ein Dashboard anzuhängen, werden die Daten der Storage-Lens-Gruppe weder erfasst noch angezeigt.

Note

- Wenn Sie eine S3-Storage-Lens-Gruppe erstellen, legen Sie eine AWS-Ressource an. Daher hat jede Storage-Lens-Gruppe ihren eigenen Amazon-Ressourcennamen (ARN), den Sie angeben können, wenn Sie sie [an ein S3-Storage-Lens-Dashboard anhängen oder daraus ausschließen](#).
- Wenn eine Storage-Lens-Gruppe nicht an ein Dashboard angehängt ist, fallen für Sie keine zusätzlichen Gebühren für die Erstellung einer Storage-Lens-Gruppe an.
- S3 Storage Lens aggregiert Nutzungsmetriken für ein Objekt aus allen passenden Storage-Lens-Gruppen. Wenn also ein Objekt den Filterbedingungen für zwei oder mehr Storage-Lens-Gruppen entspricht, werden Sie in Ihrer Speichernutzung wiederholte Angaben für dasselbe Objekt sehen.

Sie können eine Storage-Lens-Gruppe auf Kontoebene in einer bestimmten Heimatregion (aus der Liste der unterstützten AWS-Regionen) erstellen. Anschließend können Sie die Storage-Lens-Gruppe mehreren Storage-Lens-Dashboards zuordnen, sofern sich die Dashboards in demselben AWS-Konto und in derselben Heimatregion befinden. Sie sind in der Lage, pro Heimatregion in jedem AWS-Konto bis zu 50 Storage-Lens-Gruppen zu erstellen.

Die Verwaltung und Erstellung von S3-Storage-Lens-Gruppen ist mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface, (AWS CLI), der AWS-SDKs oder der Amazon-S3-REST-API möglich.

Themen

- [Anzeigen aggregierter Metriken von Storage-Lens-Gruppen](#)
- [Berechtigungen für Storage-Lens-Gruppen](#)
- [Konfiguration von Storage-Lens-Gruppen](#)
- [AWS-Ressourcen-Tags](#)
- [Metrikexport für Storage-Lens-Gruppen](#)

Anzeigen aggregierter Metriken von Storage-Lens-Gruppen

Sie können die aggregierten Metriken für Storage-Lens-Gruppen anzeigen, indem Sie die Gruppen an ein Dashboard anhängen. Die anzuhängenden Storage-Lens-Gruppen müssen sich in der angegebenen Heimatregion des Dashboard-Kontos befinden.

Um eine Storage-Lens-Gruppe an ein Dashboard anzuhängen, müssen Sie die Gruppe im Abschnitt Aggregation von Storage-Lens-Gruppen der Dashboard-Konfiguration angeben. Sind mehrere Storage-Lens-Gruppen vorhanden, können Sie die Ergebnisse der Aggregation von Storage-Lens-Gruppen filtern und bestimmte Gruppen ein- oder ausschließen. Weitere Informationen zum Anhängen von Gruppen an Dashboards finden Sie unter [the section called “Anhängen oder Entfernen einer Storage-Lens-Gruppe”](#).

Nachdem Sie die Gruppen angehängt haben, erhalten Sie innerhalb von 48 Stunden die zusätzlichen Aggregationsdaten der Storage-Lens-Gruppen im Dashboard.

Note

Wenn Sie aggregierte Metriken für die Storage-Lens-Gruppe einsehen möchten, müssen Sie die Gruppe an das S3-Storage-Lens-Dashboard anhängen.

Berechtigungen für Storage-Lens-Gruppen

Storage-Lens-Gruppen erfordern gewisse Berechtigungen in AWS Identity and Access Management (IAM), um den Zugriff auf S3-Storage-Lens-Gruppen-Aktionen zu autorisieren. Zum Erteilen dieser Berechtigungen können Sie eine identitätsbasierte IAM-Richtlinie verwenden. Die Richtlinie lässt sich an IAM-Benutzer, -Gruppen oder -Rollen anhängen, wodurch diese die entsprechenden Berechtigungen erhalten. Solche Berechtigungen können die Fähigkeit beinhalten, Storage-Lens-Gruppen zu erstellen oder zu löschen, ihre Konfigurationen einzusehen oder ihre Tags zu verwalten.

Der IAM-Benutzer oder die IAM-Rolle, dem bzw. der Sie Berechtigungen gewähren, muss zu dem Konto gehören, das die Storage-Lens-Gruppe erstellt hat oder dem diese angehört.

Zur Nutzung von Storage-Lens-Gruppen und zur Anzeige der Metriken von Storage-Lens-Gruppen benötigen Sie zunächst die entsprechenden Berechtigungen für die Verwendung von S3 Storage Lens. Weitere Informationen finden Sie unter [the section called “Berechtigungen für S3 Storage Lens”](#).

Für die Erstellung und Verwaltung von S3-Storage-Lens-Gruppen sind je nachdem, welche Aktionen Sie ausführen möchten, die folgenden IAM-Berechtigungen erforderlich:

Aktion	IAM-Berechtigungen
Erstellen einer neuen Storage-Lens-Gruppe	<code>s3:CreateStorageLensGroup</code>
Erstellen einer neuen Storage-Lens-Gruppe	<code>s3:CreateStorageLensGroup</code> , <code>s3:TagResource</code>
Aktualisieren einer vorhandenen Storage-Lens-Gruppe	<code>s3:UpdateStorageLensGroup</code>
Zurückgeben der Konfigurationsdetails einer Storage-Lens-Gruppe	<code>s3:GetStorageLensGroup</code>
Auflisten aller Storage-Lens-Gruppen in der Heimatregion	<code>s3:ListStorageLensGroups</code>
Löschen einer Storage-Lens-Gruppe	<code>s3>DeleteStorageLensGroup</code>
Auflisten der Tags, die der Storage-Lens-Gruppe hinzugefügt wurden	<code>s3:ListTagsForResource</code>

Aktion	IAM-Berechtigungen
Hinzufügen oder Aktualisieren eines Storage-Lens-Gruppen-Tags für eine vorhandene Storage-Lens-Gruppe	s3:TagResource
Löschen eines Tags aus einer Storage-Lens-Gruppe	s3:UntagResource

Im Folgenden finden Sie ein Beispiel für die Konfiguration der IAM-Richtlinie in dem Konto, in dem die Storage-Lens-Gruppe erstellt wurde. Zur Verwendung dieser Richtlinie ersetzen Sie *us-east-1* durch die Heimatregion, in der sich die Storage-Lens-Gruppe befindet. Ersetzen Sie *111122223333* durch Ihre AWS-Konto-ID und *example-storage-lens-group* durch den Namen der Storage-Lens-Gruppe. Wenn Sie diese Berechtigungen auf alle Storage-Lens-Gruppen anwenden möchten, ersetzen Sie *example-storage-lens-group* durch ein ***.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EXAMPLE-Statement-ID",
      "Effect": "Allow",
      "Action": [
        "s3:CreateStorageLensGroup",
        "s3:UpdateStorageLensGroup",
        "s3:GetStorageLensGroup",
        "s3:ListStorageLensGroups",
        "s3>DeleteStorageLensGroup",
        "s3:TagResource",
        "s3:UntagResource",
        "s3:ListTagsForResource"
      ],
      "Resource": "arn:aws:s3:us-east-1:111122223333:storage-lens-group/example-storage-lens-group"
    }
  ]
}
```

Weitere Informationen zur Verwendung von S3-Storage-Lens-Berechtigungen finden Sie unter [Berechtigungen für Amazon S3 Storage Lens](#). Weitere Informationen zur IAM-Richtliniensprache finden Sie unter [Richtlinien und Berechtigungen in Amazon S3](#).

Konfiguration von Storage-Lens-Gruppen

Namen von S3-Storage-Lens-Gruppen

Wir empfehlen, Storage-Lens-Gruppen Namen zu geben, die auf ihren Zweck hinweisen. Auf diese Weise können Sie leicht feststellen, an welche Gruppen Dashboards angehängt werden sollen. Um [eine Storage-Lens-Gruppe an ein Dashboard anzuhängen](#), müssen Sie die Gruppe im Abschnitt Aggregation von Storage-Lens-Gruppen der Dashboard-Konfiguration angeben.

Namen von Storage-Lens-Gruppen müssen innerhalb des Kontos eindeutig sein. Sie dürfen nicht länger als 64 Zeichen sein. Nur Buchstaben (a–z, A–Z), Zahlen (0–9), Bindestriche (-) und Unterstriche (_) sind zulässig.

Heimatregion

Die Heimatregion ist die AWS-Region, in der eine Storage-Lens-Gruppe erstellt und verwaltet wird. Eine Storage-Lens-Gruppe wird in derselben Heimatregion wie das Amazon-S3-Storage-Lens-Dashboard erstellt. Die Konfiguration und die Metriken der Storage-Lens-Gruppe werden ebenfalls in dieser Region gespeichert. Sie sind in der Lage, in einer Heimatregion bis zu 50 Storage-Lens-Gruppen zu erstellen.

Nachdem Sie eine Storage-Lens-Gruppe erstellt haben, können Sie die Heimatregion nicht bearbeiten.

Scope

Um Objekte in eine Storage-Lens-Gruppe aufzunehmen, müssen sie sich im Umfang des Amazon-S3-Storage-Lens-Dashboards befinden. Der Umfang des Storage-Lens-Dashboards wird durch die Buckets bestimmt, die Sie in den Dashboard-Umfang der S3-Storage-Lens-Dashboard-Konfiguration aufgenommen haben.

Sie können verschiedene Filter für Ihre Objekte verwenden, um den Umfang einer Storage-Lens-Gruppe zu definieren. Um diese Storage-Lens-Gruppenmetriken im S3-Storage-Lens-Dashboard anzuzeigen, müssen Objekte den Filtern entsprechen, die Sie in die Storage-Lens-Gruppen aufnehmen. Angenommen, in einer Storage-Lens-Gruppe werden Objekte mit dem Präfix

marketing und dem Suffix `.png` berücksichtigt, aber keine Objekte entsprechen diesen Kriterien. In diesem Fall werden in dem täglichen Metrikexport keine Metriken für diese Storage-Lens-Gruppe generiert und im Dashboard sind keine Metriken für diese Gruppe sichtbar.

Filter

Sie können die folgenden Filter in einer S3-Storage-Lens-Gruppe verwenden:

- **Präfixe** – Gibt das [Präfix](#) der berücksichtigten Objekte an. Es handelt sich um eine Zeichenfolge am Anfang des Objektschlüsselnamens. Der Wert `images` für den Filter Präfixe schließt beispielsweise Objekte mit einem der folgenden Präfixe ein: `images/`, `images-marketing` und `images/production`. Die maximale Länge eines Präfixes beträgt 512 Byte.
- **Suffixe** – Gibt das Suffix der berücksichtigten Objekte an (z. B. `.png`, `.jpeg` oder `.csv`). Die maximale Länge eines Suffixes beträgt 1 024 Byte.
- **Objekt-Tags** – Gibt die Liste der [Objekt-Tags](#) an, nach denen gefiltert werden soll. Ein Tag-Schlüssel darf nicht mehr als 128 Unicode-Zeichen beinhalten, ein Tag-Wert maximal 256 Unicode-Zeichen. Beachten Sie, dass S3-Storage-Lens-Gruppen das Objekt nur anderen Objekten zuordnen, die auch leere Tag-Werte haben, wenn das Feld Objekt-Tag-Wert leer gelassen wird.
- **Alter** – Gibt den Altersbereich der enthaltenen Objekte in Tagen an. Es werden nur Ganzzahlen unterstützt.
- **Größe** – Gibt den Objektgrößenbereich der berücksichtigten Objekte in Byte an. Es werden nur Ganzzahlen unterstützt. Der maximal zulässige Wert ist 5 TB.

Objekt-Tags von Storage-Lens-Gruppen

Sie können [eine Storage-Lens-Gruppe erstellen](#), die bis zu 10 Objekt-Tag-Filter enthält. Das folgende Beispiel enthält zwei Schlüssel-Wert-Paare für Objekt-Tags als Filter für eine Storage-Lens-Gruppe namens *Marketing-Department*. Zur Verwendung dieses Beispiels ersetzen Sie *Marketing-Department* durch den Namen Ihrer Gruppe und *object-tag-key-1*, *object-tag-value-1* usw. durch die Schlüssel-Wert-Paare der Objekt-Tags, nach denen Sie filtern möchten.

```
{
  "Name": "Marketing-Department",
  "Filter": {
    "MatchAnyTag": [
      {
        "Key": "object-tag-key-1",
        "Value": "object-tag-value-1"
      }
    ]
  }
}
```

```
    },
    {
      "Key": "object-tag-key-2",
      "Value": "object-tag-value-2"
    }
  ]
}
}
```

Logische Operatoren (**And** oder **Or**)

Um mehrere Filterbedingungen in eine Storage-Lens-Gruppe aufzunehmen, können Sie logische Operatoren (entweder **And** oder **Or**) einsetzen. Im folgenden Beispiel besitzt die Storage-Lens-Gruppe *Marketing-Department* einen **And**-Operator, der die Filter `ObjectSize`, `Prefix` und `ObjectAge` enthält. Aufgrund des **And**-Operators werden nur Objekte, die allen Filterbedingungen entsprechen, in den Umfang der Storage-Lens-Gruppe aufgenommen.

Zur Verwendung dieses Beispiels ersetzen Sie die *user input placeholders* durch die Werte, nach denen Sie filtern möchten.

```
{
  "Name": "Marketing-Department",
  "Filter": {
    "And": {
      "MatchAnyPrefix": [
        "prefix-1",
        "prefix-2",
        "prefix-3/sub-prefix-1"
      ],
      "MatchObjectAge": {
        "DaysGreaterThan": 10,
        "DaysLessThan": 60
      },
      "MatchObjectSize": {
        "BytesGreaterThan": 10,
        "BytesLessThan": 60
      }
    }
  }
}
```

Note

Wenn Sie Objekte berücksichtigen möchten, die nur einer Filterbedingung entsprechen, ersetzen Sie in diesem Beispiel den logischen Operator `And` durch den logischen Operator `Or`.

AWS-Ressourcen-Tags

Jede S3-Storage-Lens-Gruppe gilt als AWS-Ressource mit eigenem Amazon-Ressourcenname (ARN). Daher können Sie bei der Konfiguration einer Storage-Lens-Gruppe optional AWS-Ressourcen-Tags zur Gruppe hinzufügen. Es ist möglich, für jede Storage-Lens-Gruppe bis zu 50 Tags hinzuzufügen. Zum Erstellen einer Storage-Lens-Gruppe mit Tags benötigen Sie die Berechtigungen `s3:TagResource` und `s3:CreateStorageLensGroup`.

Mithilfe von AWS-Ressourcen-Tags lassen sich Ressourcen nach Abteilung, Geschäftsbereich oder Projekt kategorisieren. Dies ist sinnvoll, wenn viele Ressourcen desselben Typs vorliegen. Durch die Anwendung von Tags können Sie eine bestimmte Storage-Lens-Gruppe auf Grundlage der ihr zugewiesenen Tags schnell identifizieren. Sie können Tags auch verwenden, um Ihre Kosten zu kategorisieren und zu verfolgen.

Wenn Sie einer Storage-Lens-Gruppe ein AWS-Ressourcen-Tag hinzufügen, aktivieren Sie außerdem die [attributbasierte Zugriffskontrolle \(ABAC\)](#). ABAC ist eine Autorisierungsstrategie, die Berechtigungen basierend auf Attributen – in diesem Fall Tags – definiert. Sie können auch Bedingungen verwenden, die Ressourcen-Tags in den IAM-Richtlinien für die [Steuerung des Zugriffs auf AWS-Ressourcen](#) angeben.

Sie können Tag (Markierung)-Schlüssel und -Werte bearbeiten und Tags (Markierungen) jederzeit von einer Ressource entfernen. Beachten Sie folgende Einschränkungen:

- Bei Tag-Schlüsseln und Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.
- Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.
- Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.
- Fügen Sie keine privaten oder vertraulichen Daten in AWS-Ressourcen-Tags ein.
- Systemtags (oder Tags mit Tag-Schlüsseln, die mit `aws :` beginnen) werden nicht unterstützt.

- Die einzelnen Tag-Schlüssel dürfen maximal 128 Zeichen lang sein. Die einzelnen Tag-Werte dürfen maximal 256 Zeichen lang sein.

Metrikexport für Storage-Lens-Gruppen

Die Metriken einer S3-Storage-Lens-Gruppe sind im [Amazon-S3-Storage-Lens-Metrikexport](#) für das Dashboard enthalten, an das die Storage-Lens-Gruppe angehängt wird. Allgemeine Informationen zur Exportfunktion für Storage-Lens-Metriken finden Sie unter [Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport](#).

Der Metrikexport für Storage-Lens-Gruppen umfasst alle S3-Storage-Lens-Metriken, die im Umfang für das Dashboard enthalten sind, an das die Storage-Lens-Gruppe angehängt wurde. Der Export beinhaltet zudem zusätzliche Metrikdaten für Storage-Lens-Gruppen.

Nach Erstellung der Storage-Lens-Gruppe wird der Metrikexport täglich an den Bucket gesendet, den Sie bei der Konfiguration des Metrikexports für das Dashboard ausgewählt haben, an das die Gruppe angehängt wird. Es kann bis zu 48 Stunden dauern, bis Sie den ersten Metrikexport erhalten.

Damit Metriken im täglichen Export generiert werden, müssen Objekte den Filtern entsprechen, die Sie in die Storage-Lens-Gruppen aufnehmen. Wenn den Filtern, die Sie in die Storage-Lens-Gruppe eingebunden haben, keine Objekte entsprechen, werden keine Metriken generiert. Entspricht ein Objekt jedoch zwei oder mehr Storage-Lens-Gruppen, wird das Objekt im Metrikexport für jede Gruppe separat aufgeführt.

Sie können Metriken für Storage-Lens-Gruppen ermitteln, indem Sie in der Spalte `record_type` des Metrikexports für das Dashboard nach einem der folgenden Werte suchen:

- `STORAGE_LENS_GROUP_BUCKET`
- `STORAGE_LENS_GROUP_ACCOUNT`

In der Spalte `record_value` wird der Ressourcen-ARN für die Storage-Lens-Gruppe angezeigt (z. B. `arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-Department`).

Verwenden von Storage-Lens-Gruppen

Amazon-S3-Storage-Lens-Gruppen aggregieren Metriken mithilfe benutzerdefinierter Filter auf der Grundlage von Objektmetadaten. Sie können S3-Storage-Lens-Metriken anhand der Präfixe, der

Suffixe, der Objekt-Tags, der Objektgröße oder des Objektalters analysieren und filtern. Zudem bieten Amazon-S3-Storage-Lens-Gruppen die Möglichkeit, die Nutzung innerhalb und zwischen Amazon-S3-Buckets zu kategorisieren. Auf diese Weise erhalten Sie mehr Informationen zum S3-Speicher und können ihn besser optimieren.

Für die Visualisierung der Daten für eine Storage-Lens-Gruppe müssen Sie zunächst [Ihre Storage-Lens-Gruppe an ein S3-Storage-Lens-Dashboard anhängen](#). Wenn Sie Storage-Lens-Gruppen im Dashboard verwalten müssen, können Sie die Dashboard-Konfiguration bearbeiten. Um festzustellen, welche Storage-Lens-Gruppen sich in Ihrem Konto befinden, können Sie sie auflisten. Auf der Registerkarte Storage-Lens-Gruppen im Dashboard können Sie jederzeit überprüfen, welche Storage-Lens-Gruppen an das Dashboard angefügt sind. Wenn Sie den Umfang einer vorhandenen Storage-Lens-Gruppe überprüfen oder aktualisieren möchten, können Sie deren Details einsehen. Storage-Lens-Gruppen können auch dauerhaft gelöscht werden.

Zum Verwalten von Berechtigungen erstellen Sie benutzerdefinierte AWS-Ressourcen-Tags und fügen sie den Storage-Lens-Gruppen hinzu. Mithilfe von AWS-Ressourcen-Tags lassen sich Ressourcen nach Abteilung, Geschäftsbereich oder Projekt kategorisieren. Dies ist sinnvoll, wenn viele Ressourcen desselben Typs vorliegen. Durch die Anwendung von Tags können Sie eine bestimmte Storage-Lens-Gruppe auf Grundlage der ihr zugewiesenen Tags schnell identifizieren.

Wenn Sie einer Storage-Lens-Gruppe ein AWS-Ressourcen-Tag hinzufügen, aktivieren Sie außerdem die [attributbasierte Zugriffskontrolle \(ABAC\)](#). ABAC ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen – in diesem Fall Tags – definiert werden. Sie können auch Bedingungen verwenden, die Ressourcen-Tags in den IAM-Richtlinien für die [Steuerung des Zugriffs auf AWS-Ressourcen](#) angeben.

Themen

- [Erstellen einer Storage-Lens-Gruppe](#)
- [Anhängen von S3-Storage-Lens-Gruppen an das Dashboard und Entfernen der Gruppen](#)
- [Visualisieren von Storage Lens-Gruppendaten](#)
- [Aktualisieren einer Storage-Lens-Gruppe](#)
- [Verwalten von AWS-Ressourcen-Tags mit Storage-Lens-Gruppen](#)
- [Auflisten aller Storage-Lens-Gruppen](#)
- [Anzeigen von Details zu Storage-Lens-Gruppen](#)
- [Löschen einer Storage-Lens-Gruppe](#)

Erstellen einer Storage-Lens-Gruppe

Die folgenden Beispiele zeigen, wie Sie eine Amazon S3 Storage Lens-Gruppe mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und erstellen AWS SDK for Java.

Verwenden der S3-Konsole

So erstellen Sie eine Storage-Lens-Gruppe

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten AWS Region aus. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
4. Wählen Sie Storage-Lens-Gruppe erstellen aus.
5. Sehen Sie sich unter Allgemein Ihre Heimatregion an und geben Sie den Namen Ihrer Storage Lens-Gruppe ein.
6. Wählen Sie unter Umfang den Filter aus, der auf die Storage-Lens-Gruppe angewendet werden soll. Zum Anwenden mehrerer Filter wählen Sie die Filter und dann den logischen Operator UND oder ODER aus.
 - Wählen Sie für den Filter Präfixe die Option Präfixe aus und geben Sie eine Präfixzeichenfolge ein. Zum Hinzufügen mehrerer Präfixe wählen Sie Präfix hinzufügen aus. Zum Entfernen eines Präfixes wählen Sie neben dem zu entfernenden Präfix Entfernen aus.
 - Wählen Sie für den Filter Objekt-Tags die Option Objekt-Tags aus und geben Sie das Schlüssel-Wert-Paar für das Objekt ein. Wählen Sie dann Tag hinzufügen aus. Zum Entfernen eines vorhandenen Tags wählen Sie neben dem zu entfernenden Tag Entfernen aus.
 - Wählen Sie für den Filter Suffixe die Option Suffixe aus und geben Sie eine Suffixzeichenfolge ein. Um mehrere Suffixe hinzuzufügen, wählen Sie Suffix hinzufügen aus. Zum Entfernen eines Suffixes wählen Sie neben dem zu entfernenden Suffix Entfernen aus.
 - Geben Sie für den Filter Alter den Altersbereich des Objekts in Tagen an. Wählen Sie Mindestobjektalter festlegen aus und geben Sie das Mindestalter des Objekts ein. Wählen Sie dann Das maximale Objektalter festlegen aus und geben Sie das maximale Objektalter ein.
 - Geben Sie für den Filter Größe den Objektgrößenbereich und die Maßeinheit an. Wählen Sie Mindestobjektgröße festlegen aus und geben Sie die Mindestobjektgröße ein. Wählen Sie Die maximale Objektgröße festlegen aus und geben Sie die maximale Objektgröße ein.

7. (Optional) Fügen Sie für AWS Ressourcen-Tags das Schlüssel-Wert-Paar hinzu und wählen Sie dann Tag hinzufügen aus.
8. Wählen Sie Storage-Lens-Gruppe erstellen aus.

Verwenden Sie den AWS CLI

Mit dem folgenden AWS CLI Beispielbefehl wird eine Storage Lens-Gruppe erstellt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --storage-lens-group=file:///./marketing-department.json
```

Der folgende AWS CLI Beispielbefehl erstellt eine Storage Lens-Gruppe mit zwei AWS Ressourcen-Tags. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --storage-lens-group=file:///./marketing-department.json \  
--tags Key=k1,Value=v1 Key=k2,Value=v2
```

JSON-Beispielkonfigurationen finden Sie unter [Konfiguration von Storage-Lens-Gruppen](#).

Verwenden des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel wird eine Storage Lens-Gruppe erstellt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example – Erstellen einer Storage-Lens-Gruppe mit einem einzigen Filter

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt. Diese Gruppe besitzt einen Filter für das Objektalter, der einen Altersbereich von *30* bis *90* Tagen festlegt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;

public class CreateStorageLensGroupWithObjectAge {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupFilter objectAgeFilter = StorageLensGroupFilter.builder()
                .matchObjectAge(MatchObjectAge.builder()
                    .daysGreaterThan(30)
                    .daysLessThan(90)
                    .build())
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                .name(storageLensGroupName)
                .filter(objectAgeFilter)
                .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
                CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

Example — Erstellen einer Storage-Lens-Gruppe mit einem **AND**-Operator, der mehrere Filter umfasst

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt. Diese Gruppe verwendet den AND-Operator, um anzugeben, dass Objekte allen Filterbedingungen entsprechen müssen. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;  
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;  
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;  
import software.amazon.awssdk.services.s3control.model.S3Tag;  
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;  
import software.amazon.awssdk.services.s3control.model.StorageLensGroupAndOperator;  
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;  
  
public class CreateStorageLensGroupWithAndFilter {  
    public static void main(String[] args) {  
        String storageLensGroupName = "Marketing-Department";  
        String accountId = "111122223333";  
  
        try {  
            // Create object tags.  
            S3Tag tag1 = S3Tag.builder()  
                .key("object-tag-key-1")  
                .value("object-tag-value-1")  
                .build();  
            S3Tag tag2 = S3Tag.builder()  
                .key("object-tag-key-2")  
                .value("object-tag-value-2")  
                .build();
```

```
StorageLensGroupAndOperator andOperator =
StorageLensGroupAndOperator.builder()
    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
    .matchAnySuffix(".png", ".gif", ".jpg")
    .matchAnyTag(tag1, tag2)
    .matchObjectAge(MatchObjectAge.builder()
        .daysGreaterThan(30)
        .daysLessThan(90).build())
    .matchObjectSize(MatchObjectSize.builder()
        .bytesGreaterThan(1000L)
        .bytesLessThan(6000L).build())
    .build();

StorageLensGroupFilter andFilter = StorageLensGroupFilter.builder()
    .and(andOperator)
    .build();

StorageLensGroup storageLensGroup = StorageLensGroup.builder()
    .name(storageLensGroupName)
    .filter(andFilter)
    .build();

CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
    .storageLensGroup(storageLensGroup)
    .accountId(accountId).build();

S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_WEST_2)
    .credentialsProvider(ProfileCredentialsProvider.create())
    .build();
s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Example — Erstellen einer Storage-Lens-Gruppe mit einem **OR**-Operator, der mehrere Filter umfasst

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt. Für diese Gruppe wird ein OR-Operator verwendet, um einen Präfixfilter (*prefix-1*, *prefix-2*, *prefix3/sub-prefix-1*) oder einen Objektgrößenfilter mit einem Größenbereich zwischen *1000* Byte und *6000* Byte anzuwenden. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupOrOperator;

public class CreateStorageLensGroupWithOrFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupOrOperator orOperator =
                StorageLensGroupOrOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                    .matchObjectSize(MatchObjectSize.builder()
                        .bytesGreaterThan(1000L)
                        .bytesLessThan(6000L)
                        .build())
                    .build();

            StorageLensGroupFilter orFilter = StorageLensGroupFilter.builder()
                .or(orOperator)
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                .name(storageLensGroupName)
                .filter(orFilter)
```

```
        .build();

        CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
        .storageLensGroup(storageLensGroup)
        .accountId(accountId).build();

        S3ControlClient s3ControlClient = S3ControlClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();
        s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

Example — Erstellen Sie eine Storage Lens-Gruppe mit einem einzigen Filter und zwei AWS Ressourcen-Tags

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt, die einen Suffixfilter aufweist. In diesem Beispiel werden der Storage Lens-Gruppe außerdem zwei AWS Ressourcen-Tags hinzugefügt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.Tag;
```



```
public class CreateStorageLensGroupWithResourceTags {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create AWS resource tags.
            Tag resourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-value-1")
                .build();
            Tag resourceTag2 = Tag.builder()
                .key("resource-tag-key-2")
                .value("resource-tag-value-2")
                .build();

            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                .matchAnySuffix(".png", ".gif", ".jpg")
                .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                .name(storageLensGroupName)
                .filter(suffixFilter)
                .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
                CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .tags(resourceTag1, resourceTag2)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
        }
    }
}
```

```
        e.printStackTrace();
    }
}
}
```

JSON-Beispielkonfigurationen finden Sie unter [Konfiguration von Storage-Lens-Gruppen](#).

Anhängen von S3-Storage-Lens-Gruppen an das Dashboard und Entfernen der Gruppen

Nachdem Sie in Amazon S3 Storage Lens ein Upgrade auf das erweiterte Kontingent durchgeführt haben, können Sie eine [Storage-Lens-Gruppe](#) an das Dashboard anhängen. Sind mehrere Storage-Lens-Gruppen vorhanden, lassen sich die gewünschten Gruppen ein- oder ausschließen.

Die Storage-Lens-Gruppen müssen sich in der angegebenen Heimatregion im Dashboard-Konto befinden. Nachdem Sie eine Storage-Lens-Gruppe an das Dashboard angehängt haben, erhalten Sie innerhalb von 48 Stunden die Aggregationsdaten der zusätzlichen Storage-Lens-Gruppe im Metrikexport.

Note

Wenn Sie aggregierte Metriken für die Storage Lens-Gruppe einsehen möchten, müssen Sie sie an das Storage-Lens-Dashboard anhängen. Beispiele für JSON-Konfigurationsdateien für eine Storage Lens-Gruppe finden Sie unter [Beispiel für S3-Storage-Lens-Konfiguration mit Storage-Lens-Gruppen in JSON](#).

Anhängen einer Storage-Lens-Gruppe an ein S3-Storage-Lens-Dashboard

So hängen Sie eine Storage-Lens-Gruppe an ein Storage-Lens-Dashboard an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich unter Storage Lens die Option Dashboards aus.
3. Aktivieren Sie das Optionsfeld für das Storage-Lens-Dashboard, an das Sie eine Storage-Lens-Gruppe anhängen möchten.
4. Wählen Sie Edit (Bearbeiten) aus.
5. Unter Metrikauswahl, wählen Sie Fortschrittliche Metriken und Empfehlungen aus.
6. Wählen Sie Aggregation von Storage-Lens-Gruppen aus.

Note

Standardmäßig ist die Option Erweiterte Metriken auch ausgewählt. Sie können diese Einstellung jedoch auch deaktivieren, da sie nicht erforderlich ist, um Daten von Storage-Lens-Gruppen zu aggregieren.

7. Scrollen Sie nach unten zu Aggregation von Storage-Lens-Gruppen und geben Sie die Storage-Lens-Gruppe(n) an, die Sie in der Datenaggregation berücksichtigen oder daraus ausschließen möchten. Sie können die folgenden Filteroptionen verwenden:
 - Wenn Sie bestimmte Storage-Lens-Gruppen berücksichtigen möchten, wählen Sie Storage-Lens-Gruppen einbeziehen aus. Wählen Sie unter Zu berücksichtigende Storage-Lens-Gruppen Ihre Storage-Lens-Gruppen aus.
 - Wenn Sie alle Storage-Lens-Gruppen berücksichtigen möchten, wählen Sie Alle Storage-Lens-Gruppen in der Heimatregion in diesem Konto berücksichtigen aus.
 - Möchten Sie bestimmte Storage-Lens-Gruppen ausschließen, wählen Sie Storage-Lens-Gruppen ausschließen aus. Wählen Sie unter Auszuschließende Storage-Lens-Gruppen die Storage-Lens-Gruppen aus, die Sie ausschließen möchten.
8. Wählen Sie Save Changes. Nachdem Sie eine Storage-Lens-Gruppe an das Dashboard angehängt haben, erhalten Sie innerhalb von 48 Stunden die Aggregationsdaten der zusätzlichen Storage-Lens-Gruppe im Metrikexport.

Entfernen einer Storage-Lens-Gruppe aus einem S3-Storage-Lens-Dashboard


So entfernen Sie eine Storage-Lens-Gruppe aus einem S3-Storage-Lens-Dashboard

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich unter Storage Lens die Option Dashboards aus.
3. Aktivieren Sie das Optionsfeld für das Storage-Lens-Dashboard, aus dem Sie eine Storage-Lens-Gruppe entfernen möchten.
4. Wählen Sie Dashboard-Konfigurationen anzeigen aus.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Scrollen Sie nach unten bis zum Abschnitt Auswahl von Metriken.

- Wählen Sie unter Aggregation von Storage-Lens-Gruppen das X neben der Storage-Lens-Gruppe aus, die Sie entfernen möchten. Dadurch wird die Storage-Lens-Gruppe entfernt.

Wenn Sie alle Storage-Lens-Gruppen in das Dashboard aufgenommen haben, deaktivieren Sie das Kontrollkästchen neben Alle Storage-Lens-Gruppen in der Heimatregion in diesem Konto berücksichtigen.

- Wählen Sie Save Changes.


 Note

Es dauert bis zu 48 Stunden, bis die Konfigurationsänderungen im Dashboard angezeigt werden.

Visualisieren von Storage Lens-Gruppendaten

Sie können Storage-Lens-Gruppendaten visualisieren, indem Sie [die Gruppe an das Amazon-S3-Storage-Lens-Dashboard anhängen](#). Nachdem Sie die Storage-Lens-Gruppe in die Aggregation von Storage-Lens-Gruppen in der Dashboard-Konfiguration aufgenommen haben, kann es bis zu 48 Stunden dauern, bis die Daten der Storage-Lens-Gruppe im Dashboard angezeigt werden.

Nach Aktualisierung der Dashboard-Konfiguration werden alle neu hinzugefügten Storage-Lens-Gruppen in der Liste der verfügbaren Ressourcen auf der Registerkarte Storage-Lens-Gruppen angezeigt. Zudem können Sie die Speichernutzung auf der Registerkarte Übersicht weiter analysieren, indem Sie die Daten nach einer anderen Dimension aufteilen. Wählen Sie z. B. eines der Elemente aus, die in den drei wichtigsten Kategorien aufgeführt sind, und dann Analysieren nach, um die Daten nach einer anderen Dimension zu aufzuteilen. Sie können nicht dieselbe Dimension wie der Filter selbst anwenden.

 Note

Es ist nicht möglich, einen Storage-Lens-Gruppenfilter zusammen mit einem Präfixfilter anzuwenden oder umgekehrt. Sie können eine Storage-Lens-Gruppe auch nicht weiter analysieren, indem Sie einen Präfixfilter verwenden.

Mithilfe der Registerkarte Storage-Lens-Gruppen im Amazon-S3-Storage-Lens-Dashboard lässt sich die Datenvisualisierung für die Storage-Lens-Gruppen anzupassen, die mit dem Dashboard verknüpft

sind. Sie können entweder die Daten für einige Storage-Lens-Gruppen, die an das Dashboard angehängt sind, oder für alle Gruppen visualisieren.

Beachten Sie bei der Visualisierung von Storage-Lens-Gruppendaten im S3-Storage-Lens-Dashboard Folgendes:

- S3 Storage Lens aggregiert Nutzungsmetriken für ein Objekt aus allen passenden Storage-Lens-Gruppen. Wenn also ein Objekt den Filterbedingungen für zwei oder mehr Storage-Lens-Gruppen entspricht, werden Sie in Ihrer Speichernutzung wiederholte Zählungen für dasselbe Objekt sehen.
- Objekte müssen den Filtern entsprechen, die Sie in die Storage-Lens-Gruppen einbinden. Wenn den Filtern, die Sie in die Storage-Lens-Gruppe eingebunden haben, keine Objekte entsprechen, werden keine Metriken generiert. Um festzustellen, ob es nicht zugewiesene Objekte gibt, überprüfen Sie die Gesamtzahl der Objekte im Dashboard auf Konto- und Bucket-Ebene.

Aktualisieren einer Storage-Lens-Gruppe

In den folgenden Beispielen wird veranschaulicht, wie eine Amazon-S3-Storage-Lens-Gruppe aktualisiert wird. Sie können eine Storage-Lens-Gruppe mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java aktualisieren.

Verwenden der S3-Konsole

So aktualisieren Sie eine Storage-Lens-Gruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter Umfang die Option Bearbeiten aus.
5. Wählen Sie auf der Seite Umfang den Filter aus, der auf die Storage-Lens-Gruppe angewendet werden soll. Zum Anwenden mehrerer Filter wählen Sie die Filter und dann den logischen Operator UND oder ODER aus.
 - Wählen Sie für den Filter Präfixe die Option Präfixe aus und geben Sie eine Präfixzeichenfolge ein. Zum Hinzufügen mehrerer Präfixe wählen Sie Präfix hinzufügen aus. Zum Entfernen eines Präfixes wählen Sie neben dem zu entfernenden Präfix Entfernen aus.

- Geben Sie für den Filter Objekt-Tags das Schlüssel-Wert-Paar für das Objekt ein. Wählen Sie dann Tag hinzufügen aus. Zum Entfernen eines vorhandenen Tags wählen Sie neben dem zu entfernenden Tag Entfernen aus.
 - Wählen Sie für den Filter Suffixe die Option Suffixe aus und geben Sie eine Suffixzeichenfolge ein. Um mehrere Suffixe hinzuzufügen, wählen Sie Suffix hinzufügen aus. Zum Entfernen eines Suffixes wählen Sie neben dem zu entfernenden Suffix Entfernen aus.
 - Geben Sie für den Filter Alter den Altersbereich des Objekts in Tagen an. Wählen Sie Mindestobjektalter festlegen aus und geben Sie das Mindestalter des Objekts ein. Geben Sie für Das maximale Objektalter festlegen das maximale Objektalter ein.
 - Geben Sie für den Filter Größe den Objektgrößenbereich und die Maßeinheit an. Wählen Sie Mindestobjektgröße festlegen aus und geben Sie die Mindestobjektgröße ein. Geben Sie für Die maximale Objektgröße festlegen die maximale Objektgröße ein.
6. Wählen Sie Save Changes. Die Detailseite für die Storage-Lens-Gruppe wird angezeigt.
 7. (Optional) Wenn Sie ein neues AWS-Ressourcen-Tag hinzufügen möchten, scrollen Sie zum Abschnitt AWS-Ressourcen-Tags und wählen Sie dann Tags hinzufügen aus. Die Seite Add tags (Tags hinzufügen) wird angezeigt.

Fügen Sie das neue Schlüssel-Wert-Paar hinzu und wählen Sie dann Änderungen speichern aus. Die Detailseite für die Storage-Lens-Gruppe wird angezeigt.

8. (Optional) Wenn Sie ein vorhandenes AWS-Ressourcen-Tag entfernen möchten, scrollen Sie zum Abschnitt AWS-Ressourcen-Tags und wählen Sie das Ressourcen-Tag aus. Wählen Sie dann Löschen. Das Dialogfeld AWS-Ressourcen-Tags löschen wird angezeigt.

Wählen Sie erneut Löschen aus, um das AWS-Ressourcen-Tag dauerhaft zu löschen.

Note

Nachdem Sie ein AWS-Ressourcen-Tag dauerhaft gelöscht haben, kann es nicht wiederhergestellt werden.

Verwendung von AWS CLI

Mit dem folgenden AWS CLI-Beispielbefehl werden die Konfigurationsdetails für eine Storage-Lens-Gruppe namens *marketing-department* zurückgegeben. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --name marketing-department
```

Im folgenden AWS CLI-Beispiel wird eine Storage-Lens-Gruppe aktualisiert. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control update-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --storage-lens-group=file:///./marketing-department.json
```

JSON-Beispielkonfigurationen finden Sie unter [Konfiguration von Storage-Lens-Gruppen](#).

Verwenden des AWS-SDKs für Java

Im folgenden AWS SDK for Java-Beispiel werden die Konfigurationsdetails für die Storage-Lens-Gruppe *Marketing-Department* im Konto *111122223333* zurückgegeben. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;  
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;  
  
public class GetStorageLensGroup {  
    public static void main(String[] args) {  
        String storageLensGroupName = "Marketing-Department";  
        String accountId = "111122223333";  
  
        try {  
            GetStorageLensGroupRequest getRequest =  
                GetStorageLensGroupRequest.builder()  
                    .name(storageLensGroupName)  
                    .accountId(accountId).build();  
            S3ControlClient s3ControlClient = S3ControlClient.builder()  
                .region(Region.US_WEST_2)  
                .credentialsProvider(ProfileCredentialsProvider.create())
```

```

        .build();
        GetStorageLensGroupResponse response =
s3ControlClient.getStorageLensGroup(getRequest);
        System.out.println(response);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
}

```

In dem folgenden Beispiel wird die Storage-Lens-Gruppe *Marketing-Department* im Konto *111122223333* aktualisiert. In diesem Beispiel wird der Dashboard-Umfang so aktualisiert, dass die Objekte berücksichtigt werden, die einem der folgenden Suffixe entsprechen: *.png*, *.gif*, *.jpg* oder *.jpeg*. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```

package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.UpdateStorageLensGroupRequest;

public class UpdateStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create updated filter.
            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                .matchAnySuffix(".png", ".gif", ".jpg", ".jpeg")
                .build();

```



```
StorageLensGroup storageLensGroup = StorageLensGroup.builder()
    .name(storageLensGroupName)
    .filter(suffixFilter)
    .build();

UpdateStorageLensGroupRequest updateStorageLensGroupRequest =
UpdateStorageLensGroupRequest.builder()
    .name(storageLensGroupName)
    .storageLensGroup(storageLensGroup)
    .accountId(accountId)
    .build();

S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_WEST_2)
    .credentialsProvider(ProfileCredentialsProvider.create())
    .build();
s3ControlClient.updateStorageLensGroup(updateStorageLensGroupRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

JSON-Beispielkonfigurationen finden Sie unter [Konfiguration von Storage-Lens-Gruppen](#).

Verwalten von AWS-Ressourcen-Tags mit Storage-Lens-Gruppen

Jede Amazon-S3-Storage-Lens-Gruppe gilt als AWS-Ressource mit eigenem Amazon-Ressourcennamen (ARN). Daher können Sie bei der Konfiguration einer Storage-Lens-Gruppe optional AWS-Ressourcen-Tags zur Gruppe hinzufügen. Es ist möglich, für jede Storage-Lens-Gruppe bis zu 50 Tags hinzuzufügen. Zum Erstellen einer Storage-Lens-Gruppe mit Tags benötigen Sie die Berechtigungen `s3:TagResource` und `s3:CreateStorageLensGroup`.

Mithilfe von AWS-Ressourcen-Tags lassen sich Ressourcen nach Abteilung, Geschäftsbereich oder Projekt kategorisieren. Dies ist sinnvoll, wenn viele Ressourcen desselben Typs vorliegen. Durch die Anwendung von Tags können Sie eine bestimmte Storage-Lens-Gruppe auf Grundlage der ihr

zugewiesenen Tags schnell identifizieren. Darüber hinaus sind Sie in der Lage, Kosten mithilfe von Tags zu verfolgen und zu verteilen.

Wenn Sie einer Storage-Lens-Gruppe ein AWS-Ressourcen-Tag hinzufügen, aktivieren Sie außerdem die [attributbasierte Zugriffskontrolle \(ABAC\)](#). ABAC ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen – in diesem Fall Tags – definiert werden. Sie können auch Bedingungen verwenden, die Ressourcen-Tags in den IAM-Richtlinien für die [Steuerung des Zugriffs auf AWS-Ressourcen](#) angeben.

Sie können Tag (Markierung)-Schlüssel und -Werte bearbeiten und Tags (Markierungen) jederzeit von einer Ressource entfernen. Beachten Sie die folgenden Einschränkungen:

- Bei Tag-Schlüsseln und -Werten wird zwischen Groß- und Kleinschreibung unterschieden.
- Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.
- Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.
- Fügen Sie keine privaten oder vertraulichen Daten in AWS-Ressourcen-Tags ein.
- System-Tags (mit Tag-Schlüsseln, die mit `aws :` beginnen) werden nicht unterstützt.
- Die einzelnen Tag-Schlüssel dürfen maximal 128 Zeichen lang sein. Die einzelnen Tag-Werte dürfen maximal 256 Zeichen lang sein.

In den folgenden Beispielen wird veranschaulicht, wie AWS-Ressourcen-Tags mit Storage-Lens-Gruppen verwendet werden.

Themen

- [Hinzufügen eines AWS-Ressourcen-Tags zu einer Storage-Lens-Gruppe](#)
- [Aktualisieren von Tag-Werten einer Storage-Lens-Gruppe](#)
- [Löschen eines AWS-Ressourcen-Tags aus einer Storage-Lens-Gruppe](#)
- [Auflisten von Storage-Lens-Gruppen-Tags](#)

Hinzufügen eines AWS-Ressourcen-Tags zu einer Storage-Lens-Gruppe

In den folgenden Beispielen wird veranschaulicht, wie AWS-Ressourcen-Tags einer Amazon-S3-Storage-Lens-Gruppe hinzugefügt werden. Sie können Ressourcen-Tags mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java hinzufügen.

Verwenden der S3-Konsole

So fügen Sie ein AWS-Ressourcen-Tag zu einer Storage-Lens-Gruppe hinzu

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter AWS-Ressourcen-Tags die Option Tags hinzufügen aus.
5. Fügen Sie auf der Seite Tags hinzufügen das neue Schlüssel-Wert-Paar hinzu.

Note

Wenn Sie ein neues Tag hinzufügen, dessen Schlüssel mit dem eines vorhandenen Tags identisch ist, wird der vorherige Tag-Wert überschrieben.

6. (Optional) Um mehrere neue Tags hinzuzufügen, wählen Sie erneut Tag hinzufügen aus und fügen Sie weiterhin neue Einträge hinzu. Sie können der Storage-Lens-Gruppe bis zu 50 AWS-Ressourcen-Tags hinzufügen.
7. (Optional) Wenn Sie einen neu hinzugefügten Eintrag entfernen möchten, wählen Sie neben dem zu entfernenden Tag Entfernen aus.
8. Wählen Sie Save Changes.

Verwendung von AWS CLI

Mit dem folgenden AWS CLI-Beispielbefehl werden einer vorhandenen Storage-Lens-Gruppe namens *marketing-department* zwei Ressourcen-Tags hinzugefügt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control tag-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-  
department \  
--region us-east-1 --tags Key=k1,Value=v1 Key=k2,Value=v2
```

Verwenden des AWS-SDKs für Java

Im folgenden AWS SDK for Java-Beispiel werden einer vorhandenen Storage-Lens-Gruppe zwei AWS-Ressourcen-Tags hinzugefügt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;

public class TagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            Tag resourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-value-1")
                .build();
            Tag resourceTag2 = Tag.builder()
                .key("resource-tag-key-2")
                .value("resource-tag-value-2")
                .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                .resourceArn(resourceARN)
                .tags(resourceTag1, resourceTag2)
                .accountId(accountId)
                .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
        }
    }
}
```

```
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

Aktualisieren von Tag-Werten einer Storage-Lens-Gruppe

In den folgenden Beispielen wird veranschaulicht, wie Sie die Tag-Werte für eine Storage-Lens-Gruppe mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java aktualisieren.

Verwenden der S3-Konsole

So aktualisieren Sie AWS-Ressourcen-Tags für eine Storage-Lens-Gruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter AWS-Ressourcen-Tags das zu aktualisierende Tag aus.
5. Fügen Sie den neuen Tag-Wert hinzu. Verwenden Sie hierbei den Schlüssel, der dem Schlüssel des zu aktualisierenden Schlüssel-Wert-Paares entspricht. Klicken Sie auf das Häkchen-Symbol, um den Tag-Wert zu aktualisieren.

Note

Wenn Sie ein neues Tag hinzufügen, dessen Schlüssel mit dem eines vorhandenen Tags identisch ist, wird der vorherige Tag-Wert überschrieben.

6. (Optional) Wenn Sie neue Tags hinzufügen möchten, wählen Sie Tag hinzufügen aus, um neue Einträge hinzuzufügen. Die Seite Add tags (Tags hinzufügen) wird angezeigt.

Es ist möglich, für jede Storage-Lens-Gruppe bis zu 50 AWS-Ressourcen-Tags hinzuzufügen. Wenn Sie das Hinzufügen der neuen Tags beendet haben, klicken Sie auf Änderungen speichern.

7. (Optional) Wenn Sie einen neu hinzugefügten Eintrag entfernen möchten, wählen Sie neben dem zu entfernenden Tag Entfernen aus. Wenn Sie mit dem Entfernen der Tags fertig sind, wählen Sie Speichern aus.

Verwendung von AWS CLI

Mit dem folgenden AWS CLI-Beispielbefehl werden zwei Tag-Werte für die Storage-Lens-Gruppe namens *marketing-department* aktualisiert. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control tag-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-  
department \  
--region us-east-1 --tags Key=k1,Value=v3 Key=k2,Value=v4
```

Verwenden des AWS-SDKs für Java

In dem folgenden AWS SDK for Java-Beispiel werden zwei Storage-Lens-Gruppen-Tag-Werte aktualisiert. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.Tag;  
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;  
  
public class UpdateTagsForResource {  
    public static void main(String[] args) {  
        String resourceARN = "Resource_ARN";  
        String accountId = "111122223333";  
  
        try {  
            Tag updatedResourceTag1 = Tag.builder()  
                .key("resource-tag-key-1")  
                .value("resource-tag-updated-value-1")  
                .build();
```

```
    Tag updatedResourceTag2 = Tag.builder()
        .key("resource-tag-key-2")
        .value("resource-tag-updated-value-2")
        .build();
    TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
        .resourceArn(resourceARN)
        .tags(updatedResourceTag1, updatedResourceTag2)
        .accountId(accountId)
        .build();
    S3ControlClient s3ControlClient = S3ControlClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();
    s3ControlClient.tagResource(tagResourceRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Löschen eines AWS-Ressourcen-Tags aus einer Storage-Lens-Gruppe

In den folgenden Beispielen wird veranschaulicht, wie ein AWS-Ressourcen-Tag aus einer Storage-Lens-Gruppe gelöscht wird. Sie können Tags mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java löschen.

Verwenden der S3-Konsole

So löschen Sie ein AWS-Ressourcen-Tag aus einer Storage-Lens-Gruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter AWS-Ressourcen-Tags das zu löschende Schlüssel-Wert-Paar aus.

- Wählen Sie Delete (Löschen). Das Dialogfeld AWS-Ressourcen-Tags löschen wird angezeigt.

 Note

Wenn Tags für die Zugriffskontrolle verwendet werden, kann die Durchführung dieser Aktion Auswirkungen auf zugehörige Ressourcen haben. Nachdem Sie ein Tag dauerhaft gelöscht haben, kann es nicht wiederhergestellt werden.

- Wählen Sie Löschen aus, um das Schlüssel-Wert-Paar dauerhaft zu löschen.

Verwendung von AWS CLI

Mit dem folgenden AWS CLI-Befehl werden zwei AWS-Ressourcen-Tags aus einer vorhandenen Storage-Lens-Gruppe gelöscht. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control untag-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-  
Department \  
--region us-east-1 --tag-keys k1 k2
```

Verwenden des AWS-SDKs für Java

Im folgenden AWS SDK for Java-Beispiel werden zwei AWS-Ressourcen-Tags aus der Storage-Lens-Gruppe mit dem Amazon-Ressourcennamen (ARN) gelöscht, den Sie im Konto *111122223333* angeben. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.UntagResourceRequest;  
  
public class UntagResource {  
    public static void main(String[] args) {  
        String resourceARN = "Resource_ARN";
```



```
String accountId = "111122223333";

try {
    String tagKey1 = "resource-tag-key-1";
    String tagKey2 = "resource-tag-key-2";
    UntagResourceRequest untagResourceRequest = UntagResourceRequest.builder()
        .resourceArn(resourceARN)
        .tagKeys(tagKey1, tagKey2)
        .accountId(accountId)
        .build();
    S3ControlClient s3ControlClient = S3ControlClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();
    s3ControlClient.untagResource(untagResourceRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Auflisten von Storage-Lens-Gruppen-Tags

In den folgenden Beispielen wird veranschaulicht, wie die einer Storage-Lens-Gruppe zugeordneten AWS-Ressourcen-Tags aufgelistet werden. Sie können Tags mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java auflisten.

Verwenden der S3-Konsole

So überprüfen Sie die Liste der Tags und Tag-Werte für eine Storage-Lens-Gruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die betreffende Storage-Lens-Gruppe aus.

4. Scrollen Sie nach unten zum Abschnitt AWS-Ressourcen-Tags. Alle benutzerdefinierten AWS-Ressourcen-Tags, die der Storage-Lens-Gruppe hinzugefügt wurden, sind zusammen mit ihren Tag-Werten aufgeführt.

Verwendung von AWS CLI

Mit dem folgenden AWS CLI-Beispielbefehl werden alle Tag-Werte für die Storage-Lens-Gruppe namens *marketing-department* aufgelistet. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control list-tags-for-resource --account-id 111122223333 \  
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-  
department \  
--region us-east-1
```

Verwenden des AWS-SDKs für Java

Im folgenden AWS SDK for Java-Beispiel werden die Storage-Lens-Gruppen-Tag-Werte für den angegebenen Amazon-Ressourcennamen (ARN) der Storage-Lens-Gruppe aufgelistet. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceRequest;  
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceResponse;  
  
public class ListTagsForResource {  
    public static void main(String[] args) {  
        String resourceARN = "Resource_ARN";  
        String accountId = "111122223333";  
  
        try {  
            ListTagsForResourceRequest listTagsForResourceRequest =  
                ListTagsForResourceRequest.builder()  
                    .resourceArn(resourceARN)
```

```
        .accountId(accountId)
        .build();
    S3ControlClient s3ControlClient = S3ControlClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();
    ListTagsForResourceResponse response =
s3ControlClient.listTagsForResource(listTagsForResourceRequest);
    System.out.println(response);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it and returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Auflisten aller Storage-Lens-Gruppen

In den folgenden Beispielen wird veranschaulicht, wie alle Amazon-S3-Storage-Lens-Gruppen in einem AWS-Konto und in einer Heimatregion aufgelistet werden. Diese Beispiele zeigen, wie alle Storage-Lens-Gruppen mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java aufgelistet werden.

Verwenden der S3-Konsole

So listen Sie alle Storage-Lens-Gruppen in einem Konto und einer Heimatregion auf

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Unter Storage-Lens-Gruppen wird die Liste der Storage-Lens-Gruppen in Ihrem Konto angezeigt.

Verwendung von AWS CLI

Im folgenden AWS CLI-Beispiel werden alle Storage-Lens-Gruppen für Ihr Konto aufgelistet. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control list-storage-lens-groups --account-id 111122223333 \  
--region us-east-1
```

Verwenden des AWS-SDKs für Java

Im folgenden Beispiel für AWS SDK for Java werden alle Storage-Lens-Gruppen für das Konto *111122223333* aufgelistet. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;  
  
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3control.S3ControlClient;  
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsRequest;  
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsResponse;  
  
public class ListStorageLensGroups {  
    public static void main(String[] args) {  
        String accountId = "111122223333";  
  
        try {  
            ListStorageLensGroupsRequest listStorageLensGroupsRequest =  
ListStorageLensGroupsRequest.builder()  
                .accountId(accountId)  
                .build();  
            S3ControlClient s3ControlClient = S3ControlClient.builder()  
                .region(Region.US_WEST_2)  
                .credentialsProvider(ProfileCredentialsProvider.create())  
                .build();  
            ListStorageLensGroupsResponse response =  
s3ControlClient.listStorageLensGroups(listStorageLensGroupsRequest);  
            System.out.println(response);  
        } catch (AmazonServiceException e) {
```

```
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it and returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

Anzeigen von Details zu Storage-Lens-Gruppen

In den folgenden Beispielen wird veranschaulicht, wie die Konfigurationsdetails einer Amazon-S3-Storage-Lens-Gruppe angezeigt werden. Sie können diese Details mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java anzeigen.

Verwenden der S3-Konsole

So zeigen Sie die Konfigurationsdetails einer Storage-Lens-Gruppe an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die Optionsschaltfläche neben der betreffenden Storage-Lens-Gruppe aus.
4. Wählen Sie die Option View details aus. Jetzt können Sie die Details der Storage-Lens-Gruppe überprüfen.

Verwendung von AWS CLI

Im folgenden AWS CLI-Beispiel werden die Konfigurationsdetails für eine Storage-Lens-Gruppe zurückgegeben. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --name marketing-department
```

Verwenden des AWS-SDKs für Java

Im folgenden AWS SDK for Java-Beispiel werden die Konfigurationsdetails für die Storage-Lens-Gruppe namens *Marketing-Department* im Konto *111122223333* zurückgegeben. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            GetStorageLensGroupRequest getRequest =
                GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            GetStorageLensGroupResponse response =
                s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

Löschen einer Storage-Lens-Gruppe

In den folgenden Beispielen wird veranschaulicht, wie Sie eine Amazon-S3-Storage-Lens-Gruppe mithilfe der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und des AWS SDK for Java löschen.

Verwenden der S3-Konsole

So löschen Sie eine Storage-Lens-Gruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Aktivieren Sie unter Storage-Lens-Gruppen das Optionsfeld neben der zu löschenden Storage-Lens-Gruppe.
4. Wählen Sie Delete (Löschen). Das Dialogfeld Storage-Lens-Gruppe löschen wird angezeigt.
5. Wählen Sie erneut Löschen aus, um die Storage-Lens-Gruppe dauerhaft zu löschen.

Note

Nachdem Sie eine Storage-Lens-Gruppe gelöscht haben, kann sie nicht wiederhergestellt werden.

Verwendung von AWS CLI

In dem folgenden AWS CLI-Beispiel wird die Storage-Lens-Gruppe *marketing-department* gelöscht. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-storage-lens-group --account-id 111122223333 \  
--region us-east-1 --name marketing-department
```

Verwenden des AWS-SDKs für Java

In dem folgenden AWS SDK for Java-Beispiel wird die Storage-Lens-Gruppe *Marketing-Department* im Konto *111122223333* gelöscht. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.DeleteStorageLensGroupRequest;

public class DeleteStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            DeleteStorageLensGroupRequest deleteStorageLensGroupRequest =
DeleteStorageLensGroupRequest.builder()
                .name(storageLensGroupName)
                .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
            s3ControlClient.deleteStorageLensGroup(deleteStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```


Nachverfolgen von Amazon-S3-Anforderungen mit AWS X-Ray

AWS X-Ray erfasst Daten zu Anforderungen, die Ihre Anwendung verarbeitet. Sie können die Daten dann anzeigen und filtern, um Leistungsprobleme und Fehler in Ihrer verteilten Anwendungs- und Microservice-Architektur zu identifizieren und zu beheben. Für jede nachverfolgte Anforderung für Ihre Anwendung sehen Sie detaillierte Informationen zur Anforderung, zur Antwort und zu den Aufrufen, die Ihre Anwendung an nachgelagerte AWS-Ressourcen, Microservices, Datenbanken und HTTP-Web-APIs sendet.

Weitere Informationen finden Sie unter [Was ist AWS X-Ray?](#) im AWS X-Ray-Entwicklerhandbuch.

Themen

- [So funktioniert X-Ray mit Amazon S3](#)
- [Verfügbare Regionen](#)

So funktioniert X-Ray mit Amazon S3

AWS X-Ray unterstützt die Trace-Kontextverbreitung für Amazon S3, sodass Sie End-to-End-Anforderungen ansehen können, während sie Ihre gesamte Anwendung durchlaufen. X-Ray aggregiert die Daten, die von einzelnen Services wie Amazon S3, AWS Lambda und Amazon EC2 generiert werden, sowie die vielen Ressourcen, aus denen sich Ihre Anwendung zusammensetzt. Es bietet Ihnen einen Überblick über die Leistung Ihrer Anwendung.

Amazon S3 lässt sich in X-Ray integrieren, um den [Trace-Kontext](#) zu verbreiten und Ihnen eine Anforderungskette mit [Upstream- und Downstream-Knoten](#) bereitzustellen. Wenn ein Upstream-Service einen korrekt formatierten Trace-Header in seine S3-Anforderung einfügt, wird der Trace-Header bei der Übermittlung von Ereignisbenachrichtigungen an Downstream-Services wie Lambda, Amazon SQS und Amazon SNS von Amazon S3 übergeben. Wenn diese Services aktiv in X-Ray integriert sind, sind sie in einer Anforderungskette verknüpft, um Ihnen die vollständigen Details Ihrer Amazon S3-Anforderungen bereitzustellen.

Um X-Ray-Trace-Header über Amazon S3 zu senden, müssen Sie eine [formatierte X-Amzn-Trace-ID](#) in Ihre Anforderungen aufnehmen. Sie können den Amazon-S3-Client auch mithilfe der AWS X-Ray-SDKs instrumentieren. Eine Liste der unterstützten SDKs finden Sie in der [AWS-X-Ray-Dokumentation](#).

Service-Übersichten

In X-Ray-Service-Übersichten können Sie die Beziehungen zwischen Amazon S3 und anderen AWS-Services und -Ressourcen in Ihrer Anwendung nahezu in Echtzeit nachverfolgen. Um die End-to-End-Anforderungen in den X-Ray-Service-Übersichten einzusehen, können Sie in der X-Ray-Konsole eine Übersicht der Verbindungen zwischen Amazon S3 und anderen Services anzeigen, die Ihre Anwendung verwendet. Sie können leicht erkennen, wo eine hohe Latenz auftritt, die Verteilung der Knoten für diese Services anzeigen und dann Details zu den verschiedenen Services und Pfaden aufrufen, die die Performance der Anwendung beeinträchtigen.

X-Ray-Analysen

Ebenso können Sie die [X-Ray-Analysen-Konsole](#) nutzen, um Traces zu analysieren, Metriken wie Latenz und Fehlerraten anzusehen und [Erkenntnisse zu generieren](#), die Ihnen die Suche und Behebung von Fehlern erleichtern. In dieser Konsole werden auch Metriken wie durchschnittliche Latenz und Fehlerraten angezeigt. Weitere Informationen finden Sie unter [AWS-X-Ray-Konsole](#) im AWS-X-Ray-Entwicklerhandbuch.

Verfügbare Regionen

AWS X-Ray-Support für Amazon S3 ist in allen [AWS X-Ray-Regionen](#) verfügbar. Weitere Informationen finden Sie unter [Amazon S3 und AWS-X-Ray](#) im AWS-X-Ray-Entwicklerhandbuch.

Hosten einer statischen Website mit Amazon S3

Sie können Amazon S3 verwenden, um eine statische Website zu hosten. Auf einer statischen Website enthalten einzelne Webseiten statischen Inhalt. Sie könnten auch clientseitige Skripts enthalten.

Im Gegensatz dazu basiert eine dynamische Website auf serverseitiger Verarbeitung, einschließlich serverseitiger Skripts wie, PHPJSP, oder. ASP NET. Amazon S3 unterstützt kein serverseitiges Scripting, AWS verfügt aber über andere Ressourcen für das Hosten dynamischer Websites. Weitere Informationen zum Hosting von Websites finden Sie AWS unter [Webhosting](#).

Note

Sie können die AWS Amplify Konsole verwenden, um eine einseitige Web-App zu hosten. Die AWS Amplify -Konsole unterstützt Single-Page-Anwendungen, die mit Single-Page-Anwendungs-Frameworks (z. B. React JS, Vue JS, Angular JS und Nuxt) und statischen Websitegeneratoren (z. B. Gatsby JS, React-Static, Jekyll und Hugo) entwickelt wurden. Weitere Informationen finden Sie unter [Erste Schritte](#) im Benutzerhandbuch für die AWS Amplify -Konsole.

Amazon S3 S3-Website-Endpunkte werden nicht unterstütztHTTPS. Wenn Sie verwenden möchtenHTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird. Weitere Informationen finden Sie unter [Wie verwende ich, CloudFront um HTTPS Anfragen für meinen Amazon S3 S3-Bucket zu bearbeiten?](#) Informationen zur Verwendung HTTPS mit einer benutzerdefinierten Domain finden Sie unter [Konfiguration einer statischen Website mithilfe einer bei Route 53 registrierten benutzerdefinierten Domain](#).

Weitere Informationen zum Hosten einer statischen Website auf Amazon S3, einschließlich Anleitungen und step-by-step exemplarischer Vorgehensweisen, finden Sie in den folgenden Themen.

Themen

- [Website-Endpunkte](#)
- [Aktivieren des Website-Hostings](#)
- [Konfigurieren eines Indextdokuments](#)

- [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#)
- [Festlegen von Berechtigungen für den Website-Zugriff](#)
- [\(Optional\) Protokollieren des Webdatenverkehrs](#)
- [\(Optional\) Konfigurieren einer Webseitenumleitung](#)
- [Verwendung von ursprungsübergreifendem Ressourcenaustausch \(\) CORS](#)

Website-Endpunkte

Wenn Sie Ihren Bucket als statische Website konfigurieren, steht die Website an dem für die AWS-Region-spezifischen Website-Endpunkt des Buckets zur Verfügung. Website-Endpunkte unterscheiden sich von den Endpunkten, an die Sie Anfragen senden. REST API Weitere Informationen zu den Unterschieden zwischen den Endpunkten finden Sie unter [Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt](#).

Je nach Region weisen Ihre Amazon-S3-Website-Endpunkte eines der beiden folgenden Formate auf.

- s3-website dash (-) Region - `http://bucket-name.s3-website-Region.amazonaws.com`
- s3-website dot (.) Region - `http://bucket-name.s3-website.Region.amazonaws.com`

Diese URLs geben das Standard-Indexdokument zurück, das Sie für die Website konfigurieren. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter [Amazon-S3-Website-Endpunkte](#).

Note

Um die Sicherheit Ihrer statischen Amazon S3 S3-Websites zu erhöhen, sind die Endpunkt-Domains der Amazon S3 S3-Website (z. B. `s3-website-us-east-1.amazonaws.com` oder `s3-website.ap-south-1.amazonaws.com`) in der Liste der [öffentlichen](#) Suffixe () registriert. PSL Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host--`-Präfix zu verwenden, falls Sie jemals sensible Cookies im Domain-Namen für statische Amazon-S3-Webistes einrichten müssen. Diese Vorgehensweise hilft Ihnen dabei, Ihre Domain vor Cross-Site-Request-Forgery-Versuchen zu schützen (). CSRF Weitere Informationen finden Sie auf der [Set-Cookie](#)-Seite im Mozilla Developer Network.

Wenn Ihre Website öffentlich sein soll, müssen Sie Ihren gesamten Inhalt öffentlich lesbar machen, damit Ihre Kunden am Website-Endpunkt darauf zugreifen können. Weitere Informationen finden Sie unter [Festlegen von Berechtigungen für den Website-Zugriff](#).

Important

Endpunkte der Amazon S3 S3-Website unterstützen keine HTTPS Zugriffspunkte. Wenn Sie verwenden möchten HTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird. Weitere Informationen finden Sie unter [Wie verwende ich, CloudFront um HTTPS Anfragen für meinen Amazon S3 S3-Bucket zu bearbeiten?](#) Informationen zur Verwendung HTTPS mit einer benutzerdefinierten Domain finden Sie unter [Konfiguration einer statischen Website mithilfe einer bei Route 53 registrierten benutzerdefinierten Domain](#).

Buckets mit Zahlung durch den Auftraggeber erlauben keinen Zugriff über den Website-Endpunkt. Jede Anforderung an einen solchen Bucket erhält die Antwort 403 Access Denied (403 Zugriff verweigert). Weitere Informationen finden Sie unter [Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung](#).

Themen

- [Website-Endpunkt-Beispiele](#)
- [Hinzufügen eines DNS CNAME](#)
- [Verwenden einer benutzerdefinierten Domäne mit Route 53](#)
- [Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt](#)

Website-Endpunkt-Beispiele

Die folgenden Beispiele zeigen, wie Sie auf einen Amazon-S3-Bucket zugreifen können, der als statische Website konfiguriert ist.

Example – Anfordern eines Objekts auf Stammebene

Verwenden Sie die folgende URL Struktur, um ein bestimmtes Objekt anzufordern, das auf der Stammebene im Bucket gespeichert ist.

```
http://bucket-name.s3-website.Region.amazonaws.com/object-name
```

Im Folgenden wird beispielsweise das `photo.jpg` Objekt URL angefordert, das auf der Stammebene im Bucket gespeichert ist.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/photo.jpg
```

Example – Anfordern eines Objekts in einem Präfix

Verwenden Sie diese URL Struktur, um ein Objekt anzufordern, das in einem Ordner in Ihrem Bucket gespeichert ist.

```
http://bucket-name.s3-website.Region.amazonaws.com/folder-name/object-name
```

Im Folgenden wird URL das `docs/doc1.html` Objekt in Ihrem Bucket angefordert.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/docs/doc1.html
```

Hinzufügen eines DNS CNAME

Wenn Sie eine registrierte Domain haben, können Sie einen DNS CNAME Eintrag hinzufügen, der auf den Endpunkt der Amazon S3 S3-Website verweist. Wenn Sie die Domain beispielsweise registriert haben `www.example-bucket.com`, könnten Sie einen Bucket erstellen und einen DNS CNAME Datensatz hinzufügen `www.example-bucket.com`, der auf Folgendes verweist `www.example-bucket.com.s3-website.Region.amazonaws.com`. Alle Anfragen an `http://www.example-bucket.com` werden an `www.example-bucket.com.s3-website.Region.amazonaws.com` weitergeleitet.

Weitere Informationen finden Sie unter [Amazon S3 URLs mit CNAME Datensätzen anpassen](#).

Verwenden einer benutzerdefinierten Domäne mit Route 53

Anstatt über einen Amazon-S3-Website-Endpunkt auf die Website zuzugreifen, können Sie beispielsweise Ihre eigene Domäne verwenden, die bei Amazon Route 53 registriert ist, um Ihre Inhalte bereitzustellen, z. B. `example.com`. Sie können Amazon S3 mit Route 53 verwenden, um eine Website in der Root-Domäne zu hosten. Wenn Sie beispielsweise die Root-Domäne `example.com` haben und Ihre Website auf Amazon S3 hosten, können Ihre Website-Besucher von ihrem Browser aus auf die Seite zugreifen, indem sie `http://www.example.com` oder `http://example.com` eingeben.

Ein Beispiel-Walkthrough finden Sie unter [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#).

Hauptunterschiede zwischen einem Website-Endpoint und einem REST API Endpoint

Der Amazon-S3-Website-Endpoint ist auf den Zugriff über einen Webbrowser optimiert. In der folgenden Tabelle werden die wichtigsten Unterschiede zwischen einem REST API Endpoint und einem Website-Endpoint zusammengefasst.

Wichtiger Unterschied	RESTAPIEndpoint	Website-Endpoint
Zugriffskontrolle	Unterstützt öffentlichen und privaten Inhalt	Unterstützt nur öffentlich lesbaren Inhalt
Verarbeiten von Fehlermeldungen	Gibt eine XML -formatierte Fehlerantwort zurück	Gibt ein Dokument zurück HTML
Unterstützung einer Umleitung	Nicht zutreffend	Unterstützt Umleitungen auf Objekt- und auf Bucket-Ebene
Unterstützte Anfragen	Unterstützt alle Bucket- und Objekt-Vorgänge	Unterstützt nur GET und HEAD Anfragen für Objekte
Antworten auf GET und HEAD Anfragen im Stammverzeichnis eines Buckets	Gibt eine Liste der Objektschlüssel im Bucket zurück.	Gibt das in der Website-Konfiguration angegebene Indextokument zurück.
Unterstützung für Secure Sockets Layer (SSL)	Unterstützt SSL Verbindungen	Unterstützt keine SSL Verbindungen

Eine vollständige Liste der Amazon-S3-Endpunkte finden Sie unter [Amazon-S3-Endpunkte und -Kontingente](#) in der Allgemeine AWS-Referenz.

Aktivieren des Website-Hostings

Wenn Sie einen Bucket als statische Website konfigurieren, müssen Sie das statische Website-Hosting aktivieren, ein Indextdokument konfigurieren und Berechtigungen festlegen.

Sie können statisches Website-Hosting mit der Amazon S3 S3-Konsole RESTAPI, AWS SDKs, dem AWS CLI, oder aktivieren AWS CloudFormation.

Informationen zum Konfigurieren Ihrer Website mit einer benutzerdefinierten Domäne finden Sie unter [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#).

Verwenden der S3-Konsole

So aktivieren Sie das statische Website-Hosting

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie das Hosting statischer Websites aktivieren wollen.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Wählen Sie Use this bucket to host a website (Diesen Bucket zum Hosten einer Website verwenden).
6. Wählen Sie unter Static website hosting (Hosting statischer Websites) die Option Enable (Aktivieren) aus
7. Geben Sie unter Index document (Index-Dokument) den Dateinamen des Index-Dokuments ein, der typischerweise `index.html` ist.

Beim Namen des Indextdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss exakt mit dem Dateinamen des HTML Indextdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie Ihren Bucket für das Hosting von Websites konfigurieren, müssen Sie ein Indextdokument angeben. Amazon S3 gibt dieses Indextdokument

zurück, wenn Anfragen an die Root-Domäne oder einen der Unterordner gestellt werden.

Weitere Informationen finden Sie unter [Konfigurieren eines Indexdokuments](#).

8. Um ein eigenes benutzerdefiniertes Fehlerdokument für Fehler der Klasse 4XX bereitzustellen, geben Sie unter Fehlerdokument den Dateinamen des benutzerdefinierten Fehlerdokuments ein.

Beim Namen des Fehlerdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss genau mit dem Dateinamen des HTML Fehlerdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie kein benutzerdefiniertes Fehlerdokument angeben und ein Fehler auftritt, gibt Amazon S3 ein HTML Standard-Fehlerdokument zurück. Weitere Informationen finden Sie unter [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#).

9. (Optional) Wenn Sie erweiterte Umleitungsregeln angeben möchten, geben Sie im Feld JSONUmleitungsregeln eine Beschreibung der Regeln ein.

Beispielsweise können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten. Weitere Informationen finden Sie unter [Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen](#).

10. Wählen Sie Save Changes (Änderungen speichern).

Amazon S3 ermöglicht statisches Website-Hosting für Ihren Bucket. Unten auf der Seite sehen Sie unter Static website hosting (Hosting statischer Websites) den Website-Endpoint für Ihren Bucket.

11. Notieren Sie unter Static website hosting (Statisches Website-Hosting) den Wert für Endpoint (Endpoint).

Der Endpoint (Endpoint) ist der Amazon-S3-Website-Endpoint für Ihren Bucket. Nachdem Sie den Bucket als statische Website konfiguriert haben, können Sie diesen Endpoint verwenden, um Ihre Website zu testen.

Unter Verwendung der REST API

Weitere Informationen zum direkten Senden von REST Anfragen zur Aktivierung des statischen Website-Hostings finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- [PUTBucket-Website](#)
- [GETBucket-Webseite](#)

- [DELETEBucket-Webseite](#)

Mit der AWS SDKs

Um eine statische Website auf Amazon S3 zu hosten, konfigurieren Sie einen Amazon-S3-Bucket für ein Website-Hosting und laden dann Ihren Website-Inhalt in den Bucket hoch. Sie können den auch verwenden, AWS SDKs um die Website-Konfiguration programmgesteuert zu erstellen, zu aktualisieren und zu löschen. SDKs bieten Wrapper-Klassen rund um den Amazon S3 REST API. Wenn Ihre Anwendung dies erfordert, können Sie REST API Anfragen direkt von Ihrer Anwendung aus senden.

.NET

Das folgende Beispiel zeigt, wie Sie die AWS SDK for .NET zur Verwaltung der Website-Konfiguration für einen Bucket verwenden. Um einem Bucket eine Website-Konfiguration hinzuzufügen, geben Sie einen Bucket-Namen und eine Website-Konfiguration an. Die Website-Konfiguration muss ein Indexdokument enthalten und kann optional ein Fehlerdokument enthalten. Diese Dokumente müssen im Bucket gespeichert sein. Weitere Informationen finden Sie auf der [PUTBucket-Website](#). Weitere Informationen zur Amazon-S3-Website-Funktion finden Sie unter [Hosten einer statischen Website mit Amazon S3](#).

Das folgende C#-Code-Beispiel fügt dem angegebenen Bucket eine Website-Konfiguration hinzu. Die Konfiguration gibt die Namen für das Indexdokument und das Fehlerdokument an. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK for .NET](#) im AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class WebsiteConfigTest
    {
        private const string bucketName = "**** bucket name ****";
        private const string indexDocumentSuffix = "**** index object key ****"; //
        // For example, index.html.
    }
}
```

```
private const string errorDocument = "*** error object key ***"; // For
example, error.html.
// Specify your bucket region (an example region is shown).
private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
private static IAmazonS3 client;
public static void Main()
{
    client = new AmazonS3Client(bucketRegion);
    AddWebsiteConfigurationAsync(bucketName, indexDocumentSuffix,
errorDocument).Wait();
}

static async Task AddWebsiteConfigurationAsync(string bucketName,
string indexDocumentSuffix,
string errorDocument)
{
    try
    {
        // 1. Put the website configuration.
        PutBucketWebsiteRequest putRequest = new PutBucketWebsiteRequest()
        {
            BucketName = bucketName,
            WebsiteConfiguration = new WebsiteConfiguration()
            {
                IndexDocumentSuffix = indexDocumentSuffix,
                ErrorDocument = errorDocument
            }
        };
        PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);

        // 2. Get the website configuration.
        GetBucketWebsiteRequest getRequest = new GetBucketWebsiteRequest()
        {
            BucketName = bucketName
        };
        GetBucketWebsiteResponse getResponse = await
client.GetBucketWebsiteAsync(getRequest);
        Console.WriteLine("Index document: {0}",
getResponse.WebsiteConfiguration.IndexDocumentSuffix);
        Console.WriteLine("Error document: {0}",
getResponse.WebsiteConfiguration.ErrorDocument);
    }
}
```

```
        catch (AmazonS3Exception e)
        {
            Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
    }
}
}
```

PHP

Das folgende PHP Beispiel fügt dem angegebenen Bucket eine Website-Konfiguration hinzu. Die Methode `create_website_config` stellt explizit die Namen des Indextdokuments und des Fehlerdokuments bereit. Das Beispiel ruft auch die Website-Konfiguration ab und gibt die Antwort aus. Weitere Informationen zur Amazon-S3-Website-Funktion finden Sie unter [Hosten einer statischen Website mit Amazon S3](#).

Weitere Informationen zu AWS SDK for Ruby API finden Sie unter [AWS SDKfor Ruby — Version 2](#).

```
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Add the website configuration.
$s3->putBucketWebsite([
    'Bucket'           => $bucket,
    'WebsiteConfiguration' => [
        'IndexDocument' => ['Suffix' => 'index.html'],
        'ErrorDocument' => ['Key' => 'error.html']
    ]
]);
```

```
    ]
  ]);

  // Retrieve the website configuration.
  $result = $s3->getBucketWebsite([
    'Bucket' => $bucket
  ]);
  echo $result->getPath('IndexDocument/Suffix');

  // Delete the website configuration.
  $s3->deleteBucketWebsite([
    'Bucket' => $bucket
  ]);
```

Mit dem AWS CLI

Weitere Informationen zur Verwendung von AWS CLI , um einen S3-Bucket als statische Website zu konfigurieren, finden Sie unter [website](#) in der AWS CLI Befehlsreferenz.

Als Nächstes müssen Sie das Indextdokument konfigurieren und Berechtigungen festlegen.

Weitere Informationen finden Sie unter [Konfigurieren eines Indextdokuments](#) und [Festlegen von Berechtigungen für den Website-Zugriff](#).

Optional können Sie auch ein [Fehlerdokument](#), eine [Webdatenverkehrs-Protokollierung](#) oder eine [Weiterleitung](#) konfigurieren.

Konfigurieren eines Indextdokuments

Wenn Sie Website-Hosting aktivieren, müssen Sie auch ein Indextdokument konfigurieren und hochladen. Ein Indextdokument ist eine Webseite, die Amazon S3 zurückgibt, wenn eine Anforderung für die Root einer Website oder einen Unterordner erfolgt. Wenn ein Benutzer beispielsweise `http://www.example.com` in den Browser eingibt, fordert er keine spezifische Seite an. In diesem Fall stellt Amazon S3 das Indextdokument bereit, das manchmal auch als Standardseite bezeichnet wird.

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indextdokuments ein (z. B. `index.html`). Nachdem Sie das statische Website-Hosting für Ihren Bucket aktiviert haben, laden Sie eine HTML Datei mit dem Namen des Indextdokuments in Ihren Bucket hoch.

Der abschließende Schrägstrich auf Stammebene ist optional. URL Wenn Sie Ihre Website beispielsweise `index.html` als Indextdokument konfigurieren, wird eine der folgenden Optionen zurückgegeben. URLs `index.html`

```
http://example-bucket.s3-website.Region.amazonaws.com/  
http://example-bucket.s3-website.Region.amazonaws.com
```

Weitere Informationen zu Amazon-S3-Website-Endpunkten finden Sie unter [Website-Endpunkte](#).

Indextdokument und Ordner

In Amazon S3 ist ein Bucket ein eindimensionaler Container mit Objekten. Er bietet keine hierarchische Organisation wie das Dateisystem auf Ihrem Computer. Sie können jedoch mit Objektschlüsselnamen, die eine Ordnerstruktur implizieren, eine logische Hierarchie erstellen.

Betrachten wir beispielsweise einen Bucket mit drei Objekten, die die folgenden Schlüsselnamen besitzen. Obwohl diese nicht in einer physischen Hierarchie gespeichert sind, können Sie aus den Schlüsselnamen die folgende logische Ordnerstruktur ableiten:

- `sample1.jpg` – Das Objekt befindet sich im Stammverzeichnis des Buckets.
- `photos/2006/Jan/sample2.jpg` – Das Objekt befindet sich im Unterordner `photos/2006/Jan`.
- `photos/2006/Feb/sample3.jpg` – Das Objekt befindet sich im Unterordner `photos/2006/Feb`.

In der Amazon-S3-Konsole können Sie auch einen Ordner in einem Bucket erstellen. Sie können beispielsweise einen Ordner mit dem Namen `photos` erstellen. Sie können Objekte in den Bucket hochladen, oder in den `photos`-Ordner im Bucket. Wenn Sie dem Bucket das Objekt `sample.jpg` hinzufügen, ist der Schlüsselname `sample.jpg`. Wenn Sie das Objekt in den Ordner `photos` hochladen, ist der Objektschlüsselname `photos/sample.jpg`.

Wenn Sie eine Ordnerstruktur in Ihrem Bucket erstellen, muss es auf jeder Ebene ein Indextdokument geben. In jedem Ordner muss das Indextdokument den gleichen Namen haben, `index.html`. Wenn ein Benutzer eine URL angibt, die einer Ordnersuche ähnelt, bestimmt das Vorhandensein oder Fehlen eines abschließenden Schrägstrichs das Verhalten der Website. Im Folgenden URL wird beispielsweise mit einem abschließenden Schrägstrich das Indextdokument zurückgegeben. `photos/index.html`

```
http://bucket-name.s3-website.Region.amazonaws.com/photos/
```

Wenn Sie jedoch den abschließenden Schrägstrich aus dem vorherigen ausschließenURL, sucht Amazon S3 zunächst nach einem Objekt photos im Bucket. Wird das Objekt photos nicht gefunden, sucht es nach einem Indextdokument, photos/index.html. Wird dieses Dokument gefunden, gibt Amazon S3 eine 302 Found-Meldung zurück und verweist auf den photos/-Schlüssel. Für nachfolgende Anforderungen für photos/ gibt Amazon S3 photos/index.html zurück. Wird das Indextdokument nicht gefunden, gibt Amazon S3 einen Fehler zurück.

Konfigurieren eines Indextdokuments

Gehen Sie wie folgt vor, um ein Indextdokument mit der S3-Konsole zu konfigurieren. Sie können ein Indextdokument auch mit dem RESTAPI, dem AWS SDKs AWS CLI, dem oder konfigurieren. AWS CloudFormation

Note

In einem Bucket mit aktivierter Versionsverwaltung können Sie mehrere Kopien der Datei „index.html“ hochladen, es wird jedoch nur die neueste Version verwendet. Weitere Informationen zur Verwendung der S3-Versionsverwaltung finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indextdokuments ein (z. B. **index.html**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Indextdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie das Indextdokument

1. Erstellen Sie eine Datei index.html.

Wenn Sie noch keine index.html Datei haben, können Sie wie folgt HTML eine Datei erstellen:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>My Website Home Page</title>
</head>
<body>
```

```
<h1>Welcome to my website</h1>
<p>Now hosted on Amazon S3!</p>
</body>
</html>
```

2. Speichern Sie die Indexdatei lokal.

Der Dateiname des Indextdokuments muss genau mit dem Namen des Indextdokuments übereinstimmen, den Sie im Dialogfeld Static website hosting (Statisches Website-Hosting) eingeben. Beim Namen des Indextdokuments wird die Groß- und Kleinschreibung berücksichtigt. Wenn Sie beispielsweise im Dialogfeld Static website hosting (Statisches Website-Hosting) `index.html` als den Namen des Index document (Indextdokuments) eingeben, muss der Dateiname des Indextdokuments ebenfalls `index.html` und nicht `Index.html` lauten.

3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Indextdokuments ein (z. B. `index.html`). Weitere Informationen finden Sie unter [Aktivieren des Website-Hostings](#).

Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Indextdokument in Ihren Bucket hochzuladen:
 - Ziehen Sie die Indexdatei per Drag & Drop in das Konsolen-Bucket-Verzeichnis.
 - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-step Eine Anleitung finden Sie unter [Objekte hochladen](#).

7. (Optional) Laden Sie andere Website-Inhalte in Ihren Bucket hoch.

Als Nächstes müssen Sie Berechtigungen für den Websitezugriff festlegen. Weitere Informationen finden Sie unter [Festlegen von Berechtigungen für den Website-Zugriff](#).

Optional können Sie auch ein [Fehlerdokument](#), eine [Webdatenverkehrs-Protokollierung](#) oder eine [Weiterleitung](#) konfigurieren.

Konfigurieren eines benutzerdefinierten Fehlerdokuments

Nachdem Sie Ihren Bucket als statische Website konfiguriert haben und ein Fehler auftritt, gibt Amazon S3 ein HTML Fehlerdokument zurück. Sie können Ihren Bucket optional mit einem benutzerdefinierten Fehlerdokument konfigurieren, sodass Amazon S3 bei Auftreten eines Fehlers dieses Dokument zurückgibt.

Note

Tritt ein Fehler auf, zeigen einige Browser beim Auftreten eines Fehlers ihre eigene Fehlermeldung an und ignorieren das Fehlerdokument, das Amazon S3 zurückgibt. Wenn beispielsweise der Fehler HTTP404 Not Found auftritt, ignoriert Google Chrome möglicherweise das von Amazon S3 zurückgegebene Fehlerdokument und zeigt einen eigenen Fehler an.

Themen

- [Amazon S3 HTTP S3-Antwortcodes](#)
- [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#)

Amazon S3 HTTP S3-Antwortcodes

In der folgenden Tabelle ist die Teilmenge der HTTP Antwortcodes aufgeführt, die Amazon S3 zurückgibt, wenn ein Fehler auftritt.

HTTPFehlercode	Beschreibung
301 Moved Permanently (301 Dauerhaft verschoben)	Wenn ein Benutzer eine Anforderung direkt an den Amazon-S3-Website-Endpunkt sendet (<code>http://s3-website. <i>Region</i>.amazonaws.com/</code>), gibt Amazon S3 die Antwort 301 Moved Permanently zurück und leitet diese Anforderungen auf <code>https://aws.amazon.com/s3/</code> um.
302 Found (302 Gefunden)	Wenn Amazon S3 eine Anforderung für einen Schlüssel <code>x</code> , <code>http://<i>bucket-name</i>.s3-website. <i>Region</i>.amazonaws.com/x</code> ohne nachfolgenden Schrägstrich erhält, sucht es zuerst nach dem Objekt mit dem Schlüsselnamen <code>x</code> . Wenn das Objekt nicht gefunden wird, stellt

HTTPFehlercode	Beschreibung
304 Not Modified (304 Nicht verändert)	<p>Amazon S3 fest, dass die Anfrage für den Unterordner x vorgesehen ist, und leitet die Anfrage um, indem es einen Schrägstrich am Ende einfügt. Es gibt 302 Found zurück.</p> <p>Amazon S3 verwendet die Anforderungs-Header <code>If-Modified-Since</code> , <code>If-Unmodified-Since</code> , <code>If-Match</code> und/oder <code>If-None-Match</code> , um festzustellen, ob das angeforderte Objekt dasselbe ist, wie das, das der Client im Cache vorhält. Ist das Objekt dasselbe, gibt der Website-Endpunkt die Antwort 304 Not Modified (304 Nicht verändert) zurück.</p>
400 Malformed Request (400 Falsch formatierte Anfrage)	<p>Der Website-Endpunkt antwortet mit 400 Malformed Request (400 Falsch formatierte Anfrage), wenn ein Benutzer versucht, über einen inkorrekten regionalen Endpunkt auf einen Bucket zuzugreifen.</p>
403 Forbidden	<p>Der Website-Endpunkt antwortet mit 403 Forbidden (403 Verboten), wenn ein Benutzer auf ein Objekt zugreifen will, das nicht öffentlich lesbar ist. Der Objekteigentümer muss das Objekt mithilfe einer Bucket-Richtlinie oder einer öffentlich lesbar machenACL.</p>

HTTPFehlercode	Beschreibung
404 Not Found (404 Nicht gefunden)	<p>Der Website-Endpunkt gibt aus den folgenden Gründen 404 Not Found (404 Nicht gefunden) zurück:</p> <ul style="list-style-type: none">• Amazon S3 stellt fest, dass sich die Website auf einen Objektschlüssel bezieht, der nicht existiert. URL• Amazon S3 geht davon aus, dass die Anforderung für ein Indextdokument gilt, das nicht existiert.• Ein in der angegebenen Bucket ist URL nicht vorhanden.• Ein in der angegebenen Bucket URL existiert, ist aber nicht als Website konfiguriert. <p>Sie können ein benutzerdefiniertes Objekt erstellen, das für 404 Not Found (404 Nicht gefunden) zurückgegeben wird. Stellen Sie sicher, dass das Dokument in den als Website konfigurierten Bucket hochgeladen ist, und dass die Konfiguration für das Website-Hosting darauf ausgelegt ist, das Dokument zu verwenden.</p> <p>Informationen darüber, wie Amazon S3 das URL als Anfrage für ein Objekt oder ein Indextdokument interpretiert, finden Sie unter Konfigurieren eines Indextdokuments.</p>
500 Service Error (500 Servicefehler)	<p>Der Website-Endpunkt reagiert mit 500 Service Error (500 Servicefehler), wenn ein interner Serverfehler auftritt.</p>
503 Service Unavailable (503 Service nicht verfügbar)	<p>Der Website-Endpunkt antwortet mit 503 Service Unavailable, wenn Amazon S3 feststellt, dass Sie Ihre Anforderungsrate reduzieren müssen.</p>

Für jeden dieser Fehler gibt Amazon S3 eine vordefinierte HTML Meldung zurück. Im Folgenden finden Sie eine HTML Beispielnachricht, die bei einer Antwort vom Typ 403 Forbidden zurückgegeben wird.

403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 873CA367A51F7EC7
- HostId: DdQeZl9vkuw5luD5HKsFaTDm9KH4PZzCPRkW3igimLbTu1DiYlvXjgyd7pVxq32

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

Konfigurieren eines benutzerdefinierten Fehlerdokuments

Wenn Sie Ihren Bucket als statische Website konfigurieren, können Sie ein benutzerdefiniertes Fehlerdokument bereitstellen, das eine benutzerfreundliche Fehlermeldung und zusätzliche Hilfe enthält. Amazon S3 gibt Ihr benutzerdefiniertes Fehlerdokument nur für die HTTP 4XX-Klasse von Fehlercodes zurück.

Um ein benutzerdefiniertes Fehlerdokument über die S3-Konsole zu konfigurieren, führen Sie die folgenden Schritte aus. Sie können ein Fehlerdokument auch mit dem RESTAPI, dem AWS SDKs AWS CLI, dem oder AWS CloudFormation konfigurieren. Weitere Informationen finden Sie hier:

- [PutBucketWebsite](#) in der Amazon Simple Storage API Service-Referenz
- [AWS: :S3: :Bucket WebsiteConfiguration](#) im Benutzerhandbuch AWS CloudFormation
- [put-bucket-website](#) in der AWS CLI Befehlsreferenz

Wenn Sie das Hosting statischer Websites für Ihren Bucket aktivieren, geben Sie den Namen des Fehlerdokuments ein (z. B. **404.html**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Fehlerdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie ein Fehlerdokument

1. Erstellen Sie ein Fehlerdokument, z. B. **404.html**.

2. Speichern Sie die Fehlerdokumentdatei lokal.

Der Name des Fehlerdokuments unterscheidet zwischen Groß- und Kleinschreibung und muss genau mit dem Namen übereinstimmen, den Sie beim Aktivieren des statischen Website-Hostings eingeben. Wenn Sie beispielsweise `404.html` im Dialogfeld **Hosten einer statischen Website** als Namen des Fehlerdokuments eingeben, muss der Dateiname des Fehlerdokuments ebenfalls `404.html` lauten.

3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Fehlerdokuments ein (z. B. `404.html`). Weitere Informationen finden Sie unter [Aktivieren des Website-Hostings](#) und [Konfigurieren eines benutzerdefinierten Fehlerdokuments](#).

Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Fehlerdokument in Ihren Bucket hochzuladen:
 - Ziehen Sie die Fehlerdokumentdatei in das Konsolen-Bucket-Verzeichnis.
 - Wählen Sie **Upload (Hochladen)** und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-step Eine Anleitung finden Sie unter [Objekte hochladen](#).

Festlegen von Berechtigungen für den Website-Zugriff

Wenn Sie einen Bucket als statische Website konfigurieren und Ihre Website öffentlich sein soll, können Sie öffentlichen Lesezugriff gewähren. Um Ihren Bucket öffentlich lesbar zu machen, müssen Sie die Block-Einstellungen für den öffentlichen Zugriff für den Bucket deaktivieren und eine Bucket-Richtlinie schreiben, die öffentlichen Lesezugriff gewährt. Wenn Ihr Bucket Objekte enthält, die nicht dem Bucket-Besitzer gehören, müssen Sie möglicherweise auch eine Objektzugriffskontrollliste (ACL) hinzufügen, die allen Benutzern Lesezugriff gewährt.

Wenn Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihren Bucket nicht deaktivieren möchten, Ihre Website aber trotzdem öffentlich zugänglich sein soll, können Sie eine

CloudFront Amazon-Distribution erstellen, um Ihre statische Website bereitzustellen. Weitere Informationen finden Sie unter [Beschleunigen Sie Ihre Website mit Amazon CloudFront](#) oder [Verwenden einer CloudFront Amazon-Distribution zur Bereitstellung einer statischen Website](#) im Amazon Route 53-Entwicklerhandbuch.

Note

Wenn ein Benutzer auf dem Endpunkt der Website ein Objekt anfordert, das nicht existiert, gibt Amazon S3 einen HTTP Antwortcode zurück 404 (Not Found). Wenn das Objekt existiert, Sie aber keine Leseberechtigung dafür erteilt haben, gibt der Website-Endpunkt den HTTP Antwortcode zurück 403 (Access Denied). Der Benutzer kann aus dem Antwortcode ableiten, ob ein bestimmtes Objekt existiert. Wenn Sie dieses Verhalten nicht wünschen, sollten Sie den Website-Support für Ihren Bucket nicht aktivieren.

Themen


- [Schritt 1: Bearbeiten der S3 Block Public Access-Einstellungen](#)
- [Schritt 2: Hinzufügen einer Bucket-Richtlinie](#)
- [Objektzugriffskontrolllisten](#)

Schritt 1: Bearbeiten der S3 Block Public Access-Einstellungen

Wenn Sie einen bestehenden Bucket als statische Website konfigurieren möchten, die einen öffentlichen Zugriff bietet, müssen Sie die Block Public Access-Einstellungen für diesen Bucket bearbeiten. Möglicherweise müssen Sie auch Ihre Einstellungen für Block Public Access auf Kontoebene bearbeiten. Amazon S3 wendet die restriktivste Kombination der Block Public Access-Einstellungen auf Bucket- und Kontoebene an.

Wenn Sie beispielsweise den öffentlichen Zugriff für einen Bucket erlauben, aber den gesamten öffentlichen Zugriff auf Kontoebene blockieren, blockiert Amazon S3 den öffentlichen Zugriff auf den Bucket weiter. In diesem Szenario müssten Sie Ihre Block Public Access-Einstellungen auf Bucket- und Kontoebene bearbeiten. Weitere Informationen finden Sie unter [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#).


Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

 Warning

Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, [Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher](#) dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugriffs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#) 



Account settings for Block Public Access are currently turned on

Account settings for Block Public Access that are enabled apply even if they are disabled for this bucket.

Block *all* public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die [Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten](#), bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Schritt 2: Hinzufügen einer Bucket-Richtlinie

Um die Objekte in Ihrem Bucket öffentlich lesbar zu machen, müssen Sie eine Bucket-Richtlinie schreiben, die jedem die `s3:GetObject`-Berechtigung erteilt.

Nachdem Sie die Einstellungen für S3 Block Public Access bearbeitet haben, können Sie eine Bucket-Richtlinie hinzufügen, um öffentlichen Lesezugriff auf den Bucket zu gewähren. Wenn Sie öffentlichen Lesezugriff gewähren, kann jeder im Internet auf Ihren Bucket zugreifen.

Important

Die zuvor genannte Richtlinie ist nur ein Beispiel und erlaubt Vollzugriff auf die Inhalte Ihres Buckets. Bevor Sie mit diesem Schritt fortfahren, lesen Sie den Abschnitt [Wie kann ich die Dateien in meinem Amazon-S3-Bucket sichern?](#), um sicherzustellen, dass Sie die bewährten Methoden zum Sichern der Dateien in Ihrem S3-Bucket und die Risiken in Zusammenhang mit der Gewährung von öffentlichem Zugriff kennen.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Bucket Policy (Bucket-Richtlinie) Edit (Bearbeiten).
4. Um öffentlichen Lesezugriff auf Ihre Website zu gewähren, kopieren Sie die folgende Bucket-Richtlinie und fügen Sie sie in den Bucket policy editor (Bucket-Richtlinieneditor) ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::Bucket-Name/*"
      ]
    }
  ]
}
```

5. Aktualisieren Sie den Resource zu Ihrem Bucket-Namen.

Im obigen Beispiel für eine Bucket-Richtlinie *Bucket-Name* ist ein Platzhalter für den Bucket-Namen. Um diese Bucket-Richtlinie mit Ihrem eigenen Bucket zu verwenden, müssen Sie diesen Namen so aktualisieren, dass er mit Ihrem Bucket übereinstimmt.

6. Wählen Sie Save Changes (Änderungen speichern).

Es wird eine Meldung angezeigt, die darauf hinweist, dass die Bucket-Richtlinie erfolgreich hinzugefügt wurde.

Wenn die Fehlermeldung `Policy has invalid resource` angezeigt wird, bestätigen Sie, dass der Bucket-Name in der Bucket-Richtlinie mit Ihrem Bucket-Namen übereinstimmt. Informationen zum Hinzufügen einer Bucket-Richtlinie finden Sie unter [Wie füge ich eine S3-Bucket-Richtlinie hinzu?](#)

Wenn Sie eine Fehlermeldung erhalten und die Bucket-Richtlinie nicht speichern können, überprüfen Sie Ihr Konto und die Bucket-Einstellungen für Block Public Access, um zu bestätigen, dass Sie den öffentlichen Zugriff auf den Bucket zulassen.

Objektzugriffskontrolllisten

Sie können eine Bucket-Richtlinie verwenden, um Ihren Objekten öffentliche Leserechte zu erteilen. Die Bucket-Richtlinie gilt jedoch nur für Objekte, die sich im Besitz des Bucket-Eigentümers befinden. Wenn Ihr Bucket Objekte enthält, die nicht dem Bucket-Besitzer gehören, sollte der Bucket-Besitzer die Objektzugriffskontrollliste (ACL) verwenden, um öffentliche READ Berechtigungen für diese Objekte zu erteilen.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlich ACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern,

unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner enforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Um ein Objekt mit einem öffentlich lesbar zu machen ACL, READ erteilen Sie der `AllUsers` Gruppe die entsprechenden Berechtigungen, wie im folgenden Grant-Element dargestellt. Fügen Sie dem Objekt dieses Grant-Element hinzu ACL. Informationen zur Verwaltung finden ACLs Sie unter [Übersicht über die Zugriffskontrollliste \(ACL\)](#).

```
<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
  </Grantee>
  <Permission>READ</Permission>
</Grant>
```

(Optional) Protokollieren des Webdatenverkehrs

Sie können optional die Amazon-S3-Server-Zugriffssprotokollierung für einen Bucket aktivieren, der als statische Website konfiguriert ist. Die Server-Zugriffssprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an Ihren Bucket gestellt werden. Weitere Informationen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffssprotokollierung](#). Wenn Sie Amazon verwenden möchten, CloudFront um [Ihre Website zu beschleunigen](#), können Sie auch die CloudFront Protokollierung verwenden. Weitere Informationen finden Sie unter [Configuring and Using Access Logs](#) im Amazon CloudFront Developer Guide.

So aktivieren Sie die Server-Zugriffssprotokollierung für Ihren statischen Website-Bucket

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Erstellen Sie in derselben Region, in der Sie den Bucket erstellt haben, der als statische Website konfiguriert ist, einen Bucket für die Protokollierung, z. B. `logs.example.com`.
3. Erstellen Sie einen Ordner für die Protokolldateien der Server-Zugriffsprotokollierung (z. B. `logs`).
4. (Optional) Wenn Sie damit die Leistung Ihrer Website verbessern CloudFront möchten, erstellen Sie einen Ordner für die CloudFront Protokolldateien (z. B. `cdn`).

Weitere Informationen finden Sie unter [Beschleunigen Sie Ihre Website mit Amazon CloudFront](#).

5. Wählen Sie in der Liste Buckets den Namen Ihres Buckets.
6. Wählen Sie Properties (Eigenschaften).
7. Wählen Sie unter Server access logging (Server-Zugriffsprotokollierung) Edit (Bearbeiten).
8. Wählen Sie Enable aus.
9. Wählen Sie im Ziel-Bucket den Bucket und das Ordnerziel für die Server-Zugriffsprotokolle aus:
 - Navigieren Sie zum Ordner- und Bucket-Speicherort:
 1. Wählen Sie Browse S3 (S3 durchsuchen).
 2. Wählen Sie den Bucket-Namen und dann den Ordner mit den Protokollen aus.
 3. Wählen Sie Choose path (Pfad wählen).
 - Geben Sie den S3-Bucket-Pfad ein, z. B. **`s3://logs.example.com/logs/`**.
10. Wählen Sie Save Changes (Änderungen speichern).

In Ihrem Protokoll-Bucket können Sie jetzt auf Ihre Protokolle zugreifen. Amazon S3 schreibt Website-Zugriffsprotokolle alle zwei Stunden in Ihren Bucket zur Protokollierung.

(Optional) Konfigurieren einer Webseitenumleitung

Wenn Ihr Amazon-S3-Bucket für das statische Website-Hosting konfiguriert ist, können Sie Umleitungsregeln für Ihren Bucket oder die darin enthaltenen Objekte konfigurieren. Sie haben die folgenden Optionen für die Konfigurierung einer Umleitung.

Themen

- [Umleiten von Anforderungen für den Website-Endpunkt Ihres Buckets an einen anderen Bucket oder eine andere Domäne](#)
- [Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen](#)

- [So leiten Sie Anforderungen für ein Objekt um](#)

Umleiten von Anforderungen für den Website-Endpunkt Ihres Buckets an einen anderen Bucket oder eine andere Domäne

Sie können alle Anforderungen für einen Website-Endpunkt eines Buckets an einen anderen Host umleiten. Wenn Sie alle Anforderungen umleiten, werden alle Anforderungen an den Website-Endpunkt an den angegebenen Bucket oder die Domäne umgeleitet.

Wenn Ihre Root-Domäne zum Beispiel `example.com` ist, und Sie Anfragen sowohl für `http://example.com`, als auch für `http://www.example.com` bereitstellen möchten, müssen Sie zwei Buckets mit den Namen `example.com` und `www.example.com` erstellen. Pflegen Sie dann den Inhalt im Bucket `example.com` und konfigurieren Sie den anderen Bucket `www.example.com` so, dass alle Anforderungen an den Bucket `example.com` umgeleitet werden. Weitere Informationen finden Sie unter [Konfigurieren einer statischen Website mit einem benutzerdefinierten Domänennamen](#).

So leiten Sie Anforderungen für einen Bucket-Website-Endpunkt um

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie unter Buckets den Namen des Buckets aus, von dem Sie Anfragen umleiten möchten (z. B. `www.example.com`).
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Wählen Sie Redirect requests for an object (Anfragen für ein Objekt umleiten).
6. Geben Sie im Feld Host name (Host-Name) den Website-Endpunkt für Ihren Bucket oder Ihre benutzerdefinierte Domäne ein.

Wenn Sie Anforderungen beispielsweise zu einer Root-Domänenadresse umleiten, geben Sie **example.com**.

7. Wählen Sie unter Protocol (Protokoll) das Protokoll für die umgeleiteten Anfragen (none (kein), http oder https).

Wenn Sie kein Protokoll angeben, ist die Standardoption none (kein).

8. Wählen Sie Save Changes (Änderungen speichern).

Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen

Unter Verwendung fortschrittliche Umleitungsregeln können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten, oder abhängig von Antwortcodes. Angenommen, Sie löschen ein Objekt in Ihrem Bucket oder benennen es um. Sie können eine Weiterleitungsregel hinzufügen, die die Anfrage an ein anderes Objekt weiterleitet. Wenn Sie einen Ordner nicht zur Verfügung stellen wollen, können Sie eine Umleitungsregel hinzufügen, um die Anfrage an eine andere Webseite umzuleiten. Sie können auch eine Weiterleitungsregel hinzufügen, um Fehlerbedingungen zu verarbeiten, indem Sie Anfragen, die den Fehler zurückgeben, an eine andere Domäne weiterleiten, wenn der Fehler verarbeitet wird.

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, können Sie optional fortschrittliche Umleitungsregeln angeben. Amazon S3 hat eine Begrenzung von 50 Routingregeln pro Websitekonfiguration. Wenn Sie mehr als 50 Routingregeln benötigen, können Sie die Objektleitung verwenden. Weitere Informationen finden Sie unter [Verwenden der S3-Konsole](#).

Weitere Informationen zur Konfiguration von Routing-Regeln mithilfe von finden Sie [PutBucketWebsite](#) in der Amazon Simple Storage Service API Reference. REST API

Important

Um Umleitungsregeln in der neuen Amazon S3 S3-Konsole zu erstellen, müssen Sie JSON Beispiele für JSON finden Sie unter [Beispiele für Umleitungsregeln](#).

So konfigurieren Sie Umleitungsregeln für eine statische Website

Gehen Sie folgendermaßen vor, um Umleitungsregeln für einen Bucket hinzuzufügen, für den das statische Website-Hosting bereits aktiviert ist.

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen eines Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Geben Sie im Feld Umleitungsregeln Ihre Umleitungsregeln ein. JSON

In der S3-Konsole beschreiben Sie die Regeln mithilfe von JSON. Beispiele für JSON finden Sie unter [Beispiele für Umleitungsregeln](#). Amazon S3 hat eine Begrenzung von 50 Routingregeln pro Websitekonfiguration.

6. Wählen Sie Save Changes (Änderungen speichern).

Routingregel-Elemente

Im Folgenden finden Sie die allgemeine Syntax für die Definition der Routing-Regeln in einer Website-Konfiguration unter JSON und XML. Um Umleitungsregeln in der neuen S3-Konsole zu konfigurieren, müssen Sie Folgendes verwenden. Beispiele für JSON finden Sie unter [Beispiele für Umleitungsregeln](#).

JSON

```
[
  {
    "Condition": {
      "HttpErrorCodeReturnedEquals": "string",
      "KeyPrefixEquals": "string"
    },
    "Redirect": {
      "HostName": "string",
      "HttpRedirectCode": "string",
      "Protocol": "http|https",
      "ReplaceKeyPrefixWith": "string",
      "ReplaceKeyWith": "string"
    }
  }
]
```

Note: Redirect must each have at least one child element. You can have either ReplaceKeyPrefix with or ReplaceKeyWith but not both.

XML

```
<RoutingRules> =
  <RoutingRules>
    <RoutingRule>...</RoutingRule>
    [<RoutingRule>...</RoutingRule>
    ...]
```

```

</RoutingRules>

<RoutingRule> =
  <RoutingRule>
    [ <Condition>...</Condition> ]
    <Redirect>...</Redirect>
  </RoutingRule>

<Condition> =
  <Condition>
    [ <KeyPrefixEquals>...</KeyPrefixEquals> ]
    [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]
  </Condition>
  Note: <Condition> must have at least one child element.

<Redirect> =
  <Redirect>
    [ <HostName>...</HostName> ]
    [ <Protocol>...</Protocol> ]
    [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith> ]
    [ <ReplaceKeyWith>...</ReplaceKeyWith> ]
    [ <HttpRedirectCode>...</HttpRedirectCode> ]
  </Redirect>

```

Note: <Redirect> must have at least one child element. You can have either ReplaceKeyPrefix with or ReplaceKeyWith but not both.

In der folgenden Tabelle werden die Elemente in der Weiterleitungsregel beschrieben.

Name	Beschreibung
RoutingRules	Container für eine Sammlung von RoutingRule -Elementen.
RoutingRule	<p>Eine Regel, die eine Bedingung identifiziert, sowie die Weiterleitung, die angewendet wird, wenn die Bedingung erfüllt ist.</p> <p>Bedingung:</p> <ul style="list-style-type: none"> Ein RoutingRules -Container muss mindestens eine Weiterleitungsregel enthalten.

Name	Beschreibung
Condition	<p>Container für die Beschreibung einer Bedingung, die für die angegebene Weiterleitung erfüllt sein muss, damit sie angewendet wird. Wenn die Weiterleitungsregel keine Bedingung enthält, wird die Regel auf alle Anfragen angewendet.</p>
KeyPrefixEquals	<p>Das Präfix des Objektschlüsselnamens, von dem die Anfragen weitergeleitet werden.</p> <p><code>KeyPrefixEquals</code> ist erforderlich, wenn <code>HttpErrorCodeReturnedEquals</code> nicht angegeben ist. Wenn <code>KeyPrefixEquals</code> und <code>HttpErrorCodeReturnedEquals</code> angegeben sind, müssen beide <code>true</code> sein, damit die Bedingung erfüllt ist.</p>
HttpErrorCodeReturnedEquals	<p>Der HTTP Fehlercode, der übereinstimmen muss, damit die Umleitung angewendet wird. Wenn ein Fehler auftritt und der Fehlercode mit diesem Wert übereinstimmt, gilt die angegebene Weiterleitung.</p> <p><code>HttpErrorCodeReturnedEquals</code> ist erforderlich, wenn <code>KeyPrefixEquals</code> nicht angegeben ist. Wenn <code>KeyPrefixEquals</code> und <code>HttpErrorCodeReturnedEquals</code> angegeben sind, müssen beide <code>true</code> sein, damit die Bedingung erfüllt ist.</p>
Redirect	<p>Container-Element, das Anweisungen für die Weiterleitung der Anfrage enthält. Sie können Anfragen an einen anderen Host oder eine andere Seite umleiten, oder ein anderes Protokoll vorgeben. Eine <code>RoutingRule</code> muss ein <code>Redirect</code>-Element besitzen. Ein <code>Redirect</code>-Element muss mindestens eines der folgenden zugeordneten Elemente enthalten: <code>Protocol</code>, <code>HostName</code>, <code>ReplaceKeyPrefixWith</code>, <code>ReplaceKeyWith</code> oder <code>HttpRedirectCode</code>.</p>

Name	Beschreibung
Protocol	<p>Das Protokoll, <code>http</code> oder <code>https</code>, das im <code>Location-Header</code> verwendet werden soll, der in der Antwort zurückgegeben wird.</p> <p>Wenn eines der zugeordneten Elemente vorhanden ist, ist <code>Protocol</code> nicht erforderlich.</p>
HostName	<p>Der Hostname, der im <code>Location-Header</code> verwendet werden soll, der in der Antwort zurückgegeben wird.</p> <p>Wenn eines der zugeordneten Elemente vorhanden ist, ist <code>HostName</code> nicht erforderlich.</p>
ReplaceKeyPrefixWith	<p>Das Präfix des Objektschlüsselnamens, der den Wert von <code>KeyPrefixEquals</code> in der Umleitungsanforderung ersetzt.</p> <p>Wenn eines der zugeordneten Elemente vorhanden ist, ist <code>ReplaceKeyPrefixWith</code> nicht erforderlich. Es kann nur bereitgestellt werden, wenn <code>ReplaceKeyWith</code> nicht bereitgestellt wird.</p>
ReplaceKeyWith	<p>Der Objektschlüssel, der im <code>Location-Header</code> verwendet werden soll und in der Antwort zurückgegeben wird.</p> <p>Wenn eines der zugeordneten Elemente vorhanden ist, ist <code>ReplaceKeyWith</code> nicht erforderlich. Es kann nur bereitgestellt werden, wenn <code>ReplaceKeyPrefixWith</code> nicht bereitgestellt wird.</p>
HttpRedirectCode	<p>Der HTTP Umleitungscode, der in dem <code>Location Header</code> verwendet werden soll, der in der Antwort zurückgegeben wird.</p> <p>Wenn eines der zugeordneten Elemente vorhanden ist, ist <code>HttpRedirectCode</code> nicht erforderlich.</p>

Beispiele für Umleitungsregeln

Die folgenden Beispiele erklären häufige Weiterleitungsfälle:

Important

Um Umleitungsregeln in der neuen Amazon S3 S3-Konsole zu erstellen, müssen Sie JSON

Example 1: Weiterleitung nach der Umbenennung eines Schlüsselpräfix

Angenommen, Ihr Bucket enthält die folgenden Objekte:

- index.html
- docs/article1.html
- docs/article2.html

Jetzt wollen Sie den Ordner von docs/ in documents/ umbenennen. Nach dieser Änderung müssen Sie Anfragen für das Präfix docs/ in documents/ weiterleiten. Beispielsweise muss die Anfragen für docs/article1.html an documents/article1.html weitergeleitet werden.

In diesem Fall fügen Sie der Konfiguration der Website die folgende Routingregel hinzu.

JSON

```
[
  {
    "Condition": {
      "KeyPrefixEquals": "docs/"
    },
    "Redirect": {
      "ReplaceKeyPrefixWith": "documents/"
    }
  }
]
```

XML

```
<RoutingRules>
```

```
<RoutingRule>
  <Condition>
    <KeyPrefixEquals>docs/</KeyPrefixEquals>
  </Condition>
  <Redirect>
    <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
  </Redirect>
</RoutingRule>
</RoutingRules>
```

Example 2: Weiterleitung von Anfragen für einen gelöschten Ordner auf eine Seite

Angenommen, Sie löschen den Ordner `images/` (d. h. Sie löschen alle Objekte mit dem Schlüsselpräfix `images/`). Sie können eine Weiterleitungsregel einrichten, die Anfrage für jedes Objekt mit dem Schlüsselpräfix `images/` auf eine Seite namens `folderdeleted.html` weiterleitet.

JSON

```
[
  {
    "Condition": {
      "KeyPrefixEquals": "images/"
    },
    "Redirect": {
      "ReplaceKeyWith": "folderdeleted.html"
    }
  }
]
```

XML

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <KeyPrefixEquals>images/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

Example 3: Bei einem HTTP Fehler weiterleiten

Nehmen wir an, wenn ein angefordertes Objekt nicht gefunden wird, möchten Sie Anfragen an eine Amazon Elastic Compute Cloud (AmazonEC2) -Instance umleiten. Fügen Sie eine Umleitungsregel hinzu, sodass der Besucher der Website, wenn der HTTP Statuscode 404 (Not Found) zurückgegeben wird, zu einer EC2 Amazon-Instance weitergeleitet wird, die die Anfrage bearbeitet.

Im folgenden Beispiel wird auch das Objektschlüsselpräfix `report-404/` in die Weiterleitung eingefügt. Wenn Sie beispielsweise eine Seite anfordern `ExamplePage.html` und dies zu einem HTTP 404-Fehler führt, wird die Anfrage auf eine Seite `report-404/ExamplePage.html` auf der angegebenen EC2 Amazon-Instance umgeleitet. Wenn es keine Routing-Regel gibt und der HTTP Fehler 404 auftritt, wird das in der Konfiguration angegebene Fehlerdokument zurückgegeben.

JSON

```
[
  {
    "Condition": {
      "HttpErrorCodeReturnedEquals": "404"
    },
    "Redirect": {
      "HostName": "ec2-11-22-333-44.compute-1.amazonaws.com",
      "ReplaceKeyPrefixWith": "report-404/"
    }
  }
]
```

XML

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
    </Condition>
    <Redirect>
      <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
      <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

So leiten Sie Anforderungen für ein Objekt um

Sie können Anfragen für ein Objekt an ein anderes Objekt umleiten oder URL indem Sie den Ort der Website-Weiterleitung in den Metadaten des Objekts festlegen. Sie richten die Umleitung ein, indem Sie den Objekt-Metadaten die `x-amz-website-redirect-location`-Eigenschaft hinzufügen. In der Amazon-S3-Konsole legen Sie den Wert für Website Redirect Location (Websiteumleitungsort) in den Metadaten des Objekts fest. Wenn Sie [Amazon S3](#) verwendenAPI, legen Sie fest `x-amz-website-redirect-location`. Die Website interpretiert anschließend das Objekt als eine 301-Umleitung.

Um eine Anfrage an ein anderes Objekt umzuleiten, richten Sie den Umleitungsstandort auf den Schlüssel für das Zielobjekt ein. Um eine Anfrage an eine externe Adresse weiterzuleitenURL, legen Sie den Umleitungsort auf den gewünschten Wert URL fest. Weitere Informationen zu Objekt-Metadaten erhalten Sie unter [Systemdefinierte Objektmetadaten](#).

Wenn Sie eine Seitenumleitung einrichten, können Sie den Inhalt des Quellobjekts beibehalten oder löschen. Wenn es in Ihrem Bucket beispielsweise das Objekt `page1.html` gibt, können Sie alle Anforderungen für diese Seite an ein anderes Objekt umleiten, `page2.html`. Sie haben hierfür zwei Möglichkeiten:

- Behalten Sie den Inhalt des Objekts `page1.html` und leiten Sie Seitenanforderungen um.
- Löschen Sie den Inhalt von `page1.html` und laden Sie ein Null-Byte-Objekt mit dem Namen `page1.html` hoch, um das vorhandene Objekt zu ersetzen und Seitenanforderungen umzuleiten.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie als statische Website konfiguriert haben (z. B. `example.com`).
3. Wählen Sie unter Objects (Objekte) Ihr Objekt aus.
4. Wählen Sie Actions (Aktionen) und dann Edit metadata (Metadaten bearbeiten).
5. Wählen Sie Metadata (Metadaten) aus.
6. Wählen Sie Add Metadata (Metadaten hinzufügen) aus.
7. Wählen Sie unter Type (Typ) die Option System Defined (Systemdefiniert) aus.
8. Wählen Sie unter Key die Option `x-amz-website-redirect-location` aus.

9. Geben Sie in Value (Wert) den Schlüsselnamen des Objekts ein, zu dem Sie umleiten möchten, z. B. `/page2.html`.

Für ein anderes Objekt im selben Bucket ist das Präfix `/` im Wert erforderlich. Sie können den Wert auch auf einen externen Wert setzen URL, `http://www.example.com` z. B.

10. Wählen Sie Metadaten bearbeiten.

Mit dem REST API

Die folgenden Amazon S3 API S3-Aktionen unterstützen den `x-amz-website-redirect-location` Header in der Anfrage. Amazon S3 speichert den Header-Wert in den Objekt-Metadaten als `x-amz-website-redirect-location`.

- [PUTObjekt](#)
- [Initiieren eines mehrteiligen Uploads](#)
- [POSTObjekt](#)
- [PUTObjekt — Kopieren](#)

Ein für das Hosting von Websites konfigurierter Bucket hat sowohl den Website-Endpunkt als auch den REST Endpunkt. Eine Anfrage für eine Seite, die als 301-Umleitung konfiguriert ist, erzeugt die folgenden möglichen Ergebnisse, abhängig vom Endpunkt der Anfrage:

- Regionsspezifischer Website-Endpunkt – Amazon S3 leitet die Seitenanforderung in Übereinstimmung mit dem Wert der `x-amz-website-redirect-location`-Eigenschaft um.
- REST Endpunkt — Amazon S3 leitet die Seitenanfrage nicht weiter. Es gibt das angefragte Objekt zurück.

Weitere Informationen zu den Endpunkten finden Sie unter [Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt](#).

Wenn Sie eine Seitenumleitung einrichten, können Sie den Inhalt des Objekts beibehalten oder löschen. Nehmen wir z. B. an, dass Sie ein `page1.html`-Objekt in Ihrem Bucket haben.

- Um den Inhalt von Seitenanfragen beizubehalten `page1.html` und nur Seitenanfragen umzuleiten, können Sie eine [PUTObject-Copy-Anfrage](#) einreichen, um ein neues `page1.html` Objekt zu erstellen, das das bestehende `page1.html` Objekt als Quelle verwendet. In Ihrer Anfrage richten Sie den `x-amz-website-redirect-location`-Header ein. Nachdem die

Anforderung abgeschlossen ist, ist die Originalseite mit ihrem Inhalt unverändert, aber Amazon S3 leitet alle Anforderungen für die Seite an den von Ihnen angegebenen Umleitungsstandort um.

- Um den Inhalt des `page1.html` Objekts zu löschen und Anfragen für die Seite umzuleiten, können Sie eine PUT Objektanforderung senden, um ein Null-Byte-Objekt hochzuladen, das denselben Objektschlüssel hat: `page1.html`. In der PUT Anfrage legen Sie `x-amz-website-redirect-location` für das neue `page1.html` Objekt fest. Nachdem die Anfrage abgeschlossen ist, hat `page1.html` keinen Inhalt mehr, und Anfragen werden an den Standort umgeleitet, der in `x-amz-website-redirect-location` angegeben ist.

Wenn Sie das Objekt mithilfe der [GETObjekt-Aktion](#) zusammen mit anderen Objektmetadaten abrufen, gibt Amazon S3 den `x-amz-website-redirect-location` Header in der Antwort zurück.

Verwendung von ursprungsübergreifendem Ressourcenaustausch () CORS

Cross-Origin Resource Sharing (CORS) definiert eine Möglichkeit, wie Client-Webanwendungen, die in eine Domain geladen sind, mit Ressourcen in einer anderen Domain interagieren können. Mit CORS Support können Sie umfangreiche clientseitige Webanwendungen mit Amazon S3 erstellen und selektiv den ursprungsübergreifenden Zugriff auf Ihre Amazon S3 S3-Ressourcen zulassen.

Dieser Abschnitt bietet einen Überblick über. CORS In den Unterthemen wird beschrieben, wie Sie die Aktivierung CORS mithilfe der Amazon S3-Konsole oder programmgesteuert mithilfe von Amazon S3 und REST API AWS SDKs.

Cross-Origin Resource Sharing: Szenarien in Anwendungsfällen

Im Folgenden finden Sie Beispielszenarien für die Verwendung von. CORS

Szenario 1

Szenario 1: Angenommen, Sie hosten eine Website in einem Amazon-S3-Bucket mit dem Namen `website`, wie in [Hosten einer statischen Website mit Amazon S3](#) beschrieben. Ihre Benutzer laden den Website-Endpunkt:

```
http://website.s3-website.us-east-1.amazonaws.com
```


Jetzt möchten Sie JavaScript auf den Webseiten, die in diesem Bucket gespeichert sind, verwenden, um authentifizierte PUT Anfragen GET und Anfragen für denselben Bucket stellen zu können, indem Sie den Amazon S3 API S3-Endpunkt für den Bucket verwenden. `website.s3.us-east-1.amazonaws.com` Ein Browser würde diese Anfragen normalerweise blockieren, aber mit können CORS Sie Ihren Bucket so konfigurieren, dass Anfragen JavaScript aus verschiedenen Quellen explizit aktiviert werden. `website.s3-website.us-east-1.amazonaws.com`

Szenario 2

Angenommen, Sie möchten eine Web-Schriftart aus Ihrem S3-Bucket hosten. Auch hier benötigen Browser eine CORS Prüfung (auch Preflight-Prüfung genannt), um Webfonts laden zu können. Sie würden den Bucket, der die Web-Schriftart hostet, deshalb so konfigurieren, dass jeder Ursprung diese Anfragen machen kann.

Wie bewertet Amazon S3 die CORS Konfiguration eines Buckets?

Wenn Amazon S3 eine Preflight-Anfrage von einem Browser empfängt, bewertet es die CORS Konfiguration für den Bucket und verwendet die erste `CORSRule` Regel, die der eingehenden Browseranfrage entspricht, um eine ursprungsübergreifende Anfrage zu aktivieren. Für die Übereinstimmung mit einer Regel müssen die folgenden Bedingungen erfüllt sei:

- Der `Origin` Header in einer CORS Anfrage an Ihren Bucket muss mit den Ursprüngen im `AllowedOrigins` Element in Ihrer Konfiguration übereinstimmen. CORS
- Die HTTP Methoden, die `Access-Control-Request-Method` in einer CORS Anfrage an Ihren Bucket angegeben sind, müssen mit der oder den Methoden übereinstimmen, die in dem `AllowedMethods` Element in Ihrer CORS Konfiguration aufgeführt sind.
- Die im Header einer Pre-Flight-Anfrage aufgeführten `Access-Control-Request-Headers` Header müssen mit den Headern im `AllowedHeaders` Element in Ihrer Konfiguration übereinstimmen. CORS

Note

Die Richtlinien ACLs und gelten weiterhin, wenn Sie sie in Ihrem Bucket aktivieren CORS.

So unterstützt Object Lambda Access Point CORS

Wenn S3 Object Lambda eine Anforderung von einem Browser empfängt oder die Anforderung einen `Origin-Header` enthält, fügt S3 Object Lambda immer das Header-Feld `"AllowedOrigins": "*"` hinzu.

Weitere Informationen zur Verwendung CORS finden Sie in den folgenden Themen.

Themen

- [Elemente einer CORS Konfiguration](#)
- [Konfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen \(\) CORS](#)
- [Testen CORS](#)
- [Fehlerbehebung für CORS](#)

Elemente einer CORS Konfiguration

Um Ihren Bucket so zu konfigurieren, dass ursprungsübergreifende Anfragen zulässig sind, erstellen Sie eine CORS Konfiguration. Bei der CORS Konfiguration handelt es sich um ein Dokument mit Elementen, die die Ursprünge angeben, denen Sie den Zugriff auf Ihren Bucket gestatten, die Operationen (HTTPMethoden), die Sie für jeden Ursprung unterstützen werden, und andere vorgangsspezifische Informationen. Sie können der Konfiguration bis zu 100 Regeln hinzufügen. Sie können die CORS Konfiguration als `cors` Unterressource zum Bucket hinzufügen.

Wenn Sie CORS in der S3-Konsole konfigurieren, müssen Sie diese verwenden, um eine CORS Konfiguration JSON zu erstellen. Die neue S3-Konsole unterstützt nur JSON CORS Konfigurationen.

Weitere Informationen zur CORS Konfiguration und den darin enthaltenen Elementen finden Sie in den folgenden Themen. Anweisungen zum Hinzufügen einer CORS Konfiguration finden Sie unter [Konfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen \(\) CORS](#).

Important

In der S3-Konsole muss die CORS Konfiguration seinJSON.

Themen

- [AllowedMethods-Element](#)
- [AllowedOrigins-Element](#)

- [AllowedHeaders-Element](#)
- [ExposeHeaders-Element](#)
- [MaxAgeSeconds-Element](#)
- [Beispiele für Konfigurationen CORS](#)

AllowedMethods-Element

In der CORS Konfiguration können Sie die folgenden Werte für das AllowedMethods Element angeben.

- GET
- PUT
- POST
- DELETE
- HEAD

AllowedOrigins-Element

Im AllowedOrigins-Element geben Sie die Ursprünge an, über die Sie domänenübergreifende Anfragen erlauben möchten, z. B. `http://www.example.com`. Die Ursprungszeichenfolge darf nur ein *-Platzhalterzeichen enthalten, wie beispielsweise `http://*.example.com`. Optional können Sie * als Ursprung angeben, sodass alle Ursprünge ursprungsübergreifende Anfragen senden dürfen. Sie können auch `https` angeben, um nur sichere Ursprünge zuzulassen.

AllowedHeaders-Element

Das AllowedHeaders-Element gibt an, welche Header in einer Preflight-Anfrage durch den `Access-Control-Request-Headers`-Header erlaubt sind. Jeder Header-Name in der `Access-Control-Request-Headers` Kopfzeile muss mit einem entsprechenden Eintrag im Element übereinstimmen. Amazon S3 sendet nur die zulässigen angeforderten Header in einer Antwort. Eine Beispielliste von Headern, die in Anfragen an Amazon S3 verwendet werden können, finden Sie im APIReferenzhandbuch zu Amazon Simple Storage Service unter [Allgemeine Anforderungsheader](#).

Jede AllowedHeaders Zeichenfolge in Ihrer Konfiguration kann höchstens ein Platzhalterzeichen (*) enthalten. Beispielsweise aktiviert `<AllowedHeader>x-amz-*</AllowedHeader>` alle für Amazon spezifischen Header.

ExposeHeaders-Element

Jedes `ExposeHeader` Element identifiziert einen Header in der Antwort, auf den Kunden von ihren Anwendungen aus zugreifen können sollen (z. B. von einem JavaScript XMLHttpRequest Objekt aus). Eine Liste gängiger Amazon S3 S3-Antwortheader finden Sie unter [Common Response Headers](#) im APIReferenzhandbuch zu Amazon Simple Storage Service.

MaxAgeSeconds-Element

Das `MaxAgeSeconds` Element gibt die Zeit in Sekunden an, in der Ihr Browser die Antwort für eine Preflight-Anfrage zwischenspeichern kann, wie durch die Ressource, die HTTP Methode und den Ursprung identifiziert.

Beispiele für Konfigurationen CORS

Statt über einen Amazon-S3-Website-Endpunkt auf eine Website zuzugreifen, können Sie Ihre eigene Domäne verwenden, wie beispielsweise `example1.com`, um Ihren Inhalt bereitzustellen. Weitere Informationen zur Verwendung Ihrer eigenen Domäne finden Sie unter [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#).

Die folgende CORS Beispielkonfiguration hat drei Regeln, die als `CORSRule` Elemente angegeben sind:

- Die erste Regel erlaubt ursprungsübergreifende PUT, POST, und DELETE Anfragen vom `http://www.example1.com` Ursprung aus. Die Regel erlaubt auch, dass alle Header in einer OPTIONS Preflight-Anfrage den Header verwenden. `Access-Control-Request-Headers` Als Antwort auf OPTIONS Preflight-Anfragen gibt Amazon S3 angeforderte Header zurück.
- Die zweite Regel gestattet dieselben ursprungsübergreifenden Anfragen wie die erste Regel, aber sie bezieht sich auf einen anderen Ursprung, `http://www.example2.com`.
- Die dritte Regel erlaubt ursprungsübergreifende GET Anfragen aller Herkunft. Das Platzhalterzeichen `*` bezieht sich auf alle Ursprünge.

JSON

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
  },
]
```

```
    "AllowedMethods": [
      "PUT",
      "POST",
      "DELETE"
    ],
    "AllowedOrigins": [
      "http://www.example1.com"
    ],
    "ExposeHeaders": []
  },
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "PUT",
      "POST",
      "DELETE"
    ],
    "AllowedOrigins": [
      "http://www.example2.com"
    ],
    "ExposeHeaders": []
  },
  {
    "AllowedHeaders": [],
    "AllowedMethods": [
      "GET"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": []
  }
]
```

XML

```
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example1.com</AllowedOrigin>

    <AllowedMethod>PUT</AllowedMethod>
```

```

    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>

    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
<CORSRule>
  <AllowedOrigin>http://www.example2.com</AllowedOrigin>

  <AllowedMethod>PUT</AllowedMethod>
  <AllowedMethod>POST</AllowedMethod>
  <AllowedMethod>DELETE</AllowedMethod>

  <AllowedHeader>*</AllowedHeader>
</CORSRule>
<CORSRule>
  <AllowedOrigin>*</AllowedOrigin>
  <AllowedMethod>GET</AllowedMethod>
</CORSRule>
</CORSConfiguration>

```

Die CORS Konfiguration erlaubt auch optionale Konfigurationsparameter, wie in der folgenden CORS Konfiguration gezeigt. In diesem Beispiel erlaubt die CORS Konfiguration ursprungsübergreifende PUT, POST, und DELETE Anfragen vom `http://www.example.com` Ursprung aus.

JSON

```

[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "PUT",
      "POST",
      "DELETE"
    ],
    "AllowedOrigins": [
      "http://www.example.com"
    ],
    "ExposeHeaders": [
      "x-amz-server-side-encryption",

```

```

        "x-amz-request-id",
        "x-amz-id-2"
    ],
    "MaxAgeSeconds": 3000
}
]

```

XML

```

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>x-amz-server-side-encryption</
ExposeHeader>
    <ExposeHeader>x-amz-request-id</
ExposeHeader>
    <ExposeHeader>x-amz-id-2</ExposeHeader>
  </CORSRule>
</CORSConfiguration>

```

Das `CORSRule`-Element in der obigen Konfiguration beinhaltet die folgenden optionalen Elemente:

- `MaxAgeSeconds`— Gibt die Zeitspanne in Sekunden (in diesem Beispiel 3000) an, für die der Browser eine Amazon S3 S3-Antwort auf eine `OPTIONS` Preflight-Anfrage für die angegebene Ressource zwischenspeichert. Durch die Zwischenspeicherung der Antwort muss der Browser keine Preflight-Anfragen an Amazon S3 senden, wenn die ursprüngliche Anfrage wiederholt werden soll.
- `ExposeHeader`— Identifiziert die Antwort-Header (in diesem Beispiel, und `x-amz-id-2`) `x-amz-server-side-encryption` `x-amz-request-id`, auf die Kunden von ihren Anwendungen aus zugreifen können (z. B. von einem Objekt aus). `JavaScript XMLHttpRequest`

Konfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen () CORS

Cross-Origin Resource Sharing (CORS) definiert eine Methode, mit der Client-Webanwendungen, die in eine Domain geladen sind, mit Ressourcen in einer anderen Domain interagieren können. Mit CORS Support können Sie umfangreiche clientseitige Webanwendungen mit Amazon S3 erstellen und selektiv den ursprungsübergreifenden Zugriff auf Ihre Amazon S3 S3-Ressourcen zulassen.

In diesem Abschnitt erfahren Sie, wie Sie CORS mithilfe der Amazon S3-Konsole RESTAPI, Amazon S3 und AWS SDKs. Um Ihren Bucket so zu konfigurieren, dass ursprungsübergreifende Anfragen zulässig sind, fügen Sie dem Bucket eine CORS Konfiguration hinzu. Eine CORS Konfiguration ist ein Dokument, das Regeln definiert, die die Ursprünge, denen Sie Zugriff auf Ihren Bucket gewähren, die für jeden Ursprung unterstützten Operationen (HTTPMethoden) und andere vorgangsspezifische Informationen festlegen. In der S3-Konsole muss es sich bei der CORS Konfiguration um ein JSON Dokument handeln.

CORSKonfigurationen in JSON und finden Sie XML beispielsweise unter [Elemente einer CORS Konfiguration](#).

Verwenden der S3-Konsole

In diesem Abschnitt wird erklärt, wie Sie die Amazon S3 S3-Konsole verwenden, um einem S3-Bucket eine Cross-Origin-Konfiguration für die gemeinsame Nutzung von Ressourcen (CORS) hinzuzufügen.

Wenn Sie die Option für CORS den Bucket aktivieren, gelten weiterhin die Zugriffskontrolllisten (ACLs) und andere Richtlinien für Zugriffsberechtigungen.

Important

In der S3-Konsole muss die CORS Konfiguration seinJSON. Beispiele für CORS Konfigurationen in JSON und XML finden Sie unter [Elemente einer CORS Konfiguration](#).


Um eine CORS Konfiguration zu einem S3-Bucket hinzuzufügen

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Bucket-Richtlinie erstellen wollen.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie im Abschnitt Cross-Origin Resource Sharing (CORS) die Option Bearbeiten aus.
5. Geben Sie im Textfeld des CORSKonfigurationseditors eine neue CORS Konfiguration ein oder kopieren Sie sie und fügen Sie sie ein, oder bearbeiten Sie eine vorhandene Konfiguration.

Die CORS Konfiguration ist eine JSON Datei. Der Text, den Sie in den Editor eingeben, muss gültig seinJSON. Weitere Informationen finden Sie unter [Elemente einer CORS Konfiguration](#).

6. Wählen Sie Save Changes (Änderungen speichern).

 Note

Amazon S3 zeigt den Amazon-Ressourcennamen (ARN) für den Bucket neben dem Titel des CORSKonfigurationseditors an. Weitere Informationen zu finden Sie ARNs unter [Amazon Resource Names \(ARNs\) und AWS Service-Namespaces](#) im Allgemeine Amazon Web Services-Referenz.

Verwendung der AWS SDKs

Sie können das AWS SDKum die ursprungsübergreifende gemeinsame Nutzung von Ressourcen (CORS) für einen Bucket zu verwalten. Weitere Informationen zu finden Sie CORS unter [Verwendung von ursprungsübergreifendem Ressourcenaustausch \(\) CORS](#).

Im Folgenden sind einige Beispiele aufgeführt:

- Erstellt eine CORS Konfiguration und legt die Konfiguration für einen Bucket fest
- Ruft die Konfiguration ab und ändert sie durch Hinzufügen einer Regel ab
- Fügt die abgeänderte Konfiguration dem Bucket hinzu
- Löscht die Konfiguration

Java

Example

Example

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketCrossOriginConfiguration;
import com.amazonaws.services.s3.model.CORSRule;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class CORS {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "**** Bucket name ****";

        // Create two CORS rules.
        List<CORSRule.AllowedMethods> rule1AM = new
        ArrayList<CORSRule.AllowedMethods>();
        rule1AM.add(CORSRule.AllowedMethods.PUT);
        rule1AM.add(CORSRule.AllowedMethods.POST);
        rule1AM.add(CORSRule.AllowedMethods.DELETE);
        CORSRule rule1 = new
        CORSRule().withId("CORSRule1").withAllowedMethods(rule1AM)
                .withAllowedOrigins(Arrays.asList("http://*.example.com"));

        List<CORSRule.AllowedMethods> rule2AM = new
        ArrayList<CORSRule.AllowedMethods>();
        rule2AM.add(CORSRule.AllowedMethods.GET);
```

```
CORSRule rule2 = new
CORSRule().withId("CORSRule2").withAllowedMethods(rule2AM)
    .withAllowedOrigins(Arrays.asList("*")).withMaxAgeSeconds(3000)
    .withExposedHeaders(Arrays.asList("x-amz-server-side-encryption"));

List<CORSRule> rules = new ArrayList<CORSRule>();
rules.add(rule1);
rules.add(rule2);

// Add the rules to a new CORS configuration.
BucketCrossOriginConfiguration configuration = new
BucketCrossOriginConfiguration();
configuration.setRules(rules);

try {
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(clientRegion)
        .build();

    // Add the configuration to the bucket.
    s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

    // Retrieve and display the configuration.
    configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
    printCORSConfiguration(configuration);

    // Add another new rule.
    List<CORSRule.AllowedMethods> rule3AM = new
ArrayList<CORSRule.AllowedMethods>();
    rule3AM.add(CORSRule.AllowedMethods.HEAD);
    CORSRule rule3 = new
CORSRule().withId("CORSRule3").withAllowedMethods(rule3AM)
        .withAllowedOrigins(Arrays.asList("http://www.example.com"));

    rules = configuration.getRules();
    rules.add(rule3);
    configuration.setRules(rules);
    s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

    // Verify that the new rule was added by checking the number of rules in
the
    // configuration.
    configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
```

```
        System.out.println("Expected # of rules = 3, found " +
configuration.getRules().size());

        // Delete the configuration.
        s3Client.deleteBucketCrossOriginConfiguration(bucketName);
        System.out.println("Removed CORS configuration.");

        // Retrieve and display the configuration to verify that it was
        // successfully deleted.
        configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
        printCORSConfiguration(configuration);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}

private static void printCORSConfiguration(BucketCrossOriginConfiguration
configuration) {
    if (configuration == null) {
        System.out.println("Configuration is null.");
    } else {
        System.out.println("Configuration has " +
configuration.getRules().size() + " rules\n");

        for (CORSRule rule : configuration.getRules()) {
            System.out.println("Rule ID: " + rule.getId());
            System.out.println("MaxAgeSeconds: " + rule.getMaxAgeSeconds());
            System.out.println("AllowedMethod: " + rule.getAllowedMethods());
            System.out.println("AllowedOrigins: " + rule.getAllowedOrigins());
            System.out.println("AllowedHeaders: " + rule.getAllowedHeaders());
            System.out.println("ExposeHeader: " + rule.getExposedHeaders());
            System.out.println();
        }
    }
}
}
```

.NET

Example

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CORSTest
    {
        private const string bucketName = "**** bucket name ****";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CORSConfigTestAsync().Wait();
        }
        private static async Task CORSConfigTestAsync()
        {
            try
            {
                // Create a new configuration request and add two rules
                CORSConfiguration configuration = new CORSConfiguration
                {
                    Rules = new System.Collections.Generic.List<CORSRule>
                    {
                        new CORSRule
                        {
                            Id = "CORSRule1",
                            AllowedMethods = new List<string> {"PUT", "POST",
"DELETE"}},
```

```
        AllowedOrigins = new List<string> {"http://
*.example.com"}
    },
    new CORSRule
    {
        Id = "CORSRule2",
        AllowedMethods = new List<string> {"GET"},
        AllowedOrigins = new List<string> {"*"},
        MaxAgeSeconds = 3000,
        ExposeHeaders = new List<string> {"x-amz-server-side-
encryption"}
    }
};

// Add the configuration to the bucket.
await PutCORSCONfigurationAsync(configuration);

// Retrieve an existing configuration.
configuration = await RetrieveCORSCONfigurationAsync();

// Add a new rule.
configuration.Rules.Add(new CORSRule
{
    Id = "CORSRule3",
    AllowedMethods = new List<string> { "HEAD" },
    AllowedOrigins = new List<string> { "http://www.example.com" }
});

// Add the configuration to the bucket.
await PutCORSCONfigurationAsync(configuration);

// Verify that there are now three rules.
configuration = await RetrieveCORSCONfigurationAsync();
Console.WriteLine();
Console.WriteLine("Expected # of rulest=3; found:{0}",
configuration.Rules.Count);
Console.WriteLine();
Console.WriteLine("Pause before configuration delete. To continue,
click Enter...");
Console.ReadKey();

// Delete the configuration.
await DeleteCORSCONfigurationAsync();
```

```
        // Retrieve a nonexistent configuration.
        configuration = await RetrieveCORSConfigurationAsync();
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
}

static async Task PutCORSConfigurationAsync(CORSConfiguration configuration)
{
    PutCORSConfigurationRequest request = new PutCORSConfigurationRequest
    {
        BucketName = bucketName,
        Configuration = configuration
    };

    var response = await s3Client.PutCORSConfigurationAsync(request);
}

static async Task<CORSConfiguration> RetrieveCORSConfigurationAsync()
{
    GetCORSConfigurationRequest request = new GetCORSConfigurationRequest
    {
        BucketName = bucketName
    };

    var response = await s3Client.GetCORSConfigurationAsync(request);
    var configuration = response.Configuration;
    PrintCORSRules(configuration);
    return configuration;
}

static async Task DeleteCORSConfigurationAsync()
{

```

```
        DeleteCORSConfigurationRequest request = new
DeleteCORSConfigurationRequest
    {
        BucketName = bucketName
    };
    await s3Client.DeleteCORSConfigurationAsync(request);
}

static void PrintCORSRules(CORSConfiguration configuration)
{
    Console.WriteLine();

    if (configuration == null)
    {
        Console.WriteLine("\nConfiguration is null");
        return;
    }

    Console.WriteLine("Configuration has {0} rules:",
configuration.Rules.Count);
    foreach (CORSRule rule in configuration.Rules)
    {
        Console.WriteLine("Rule ID: {0}", rule.Id);
        Console.WriteLine("MaxAgeSeconds: {0}", rule.MaxAgeSeconds);
        Console.WriteLine("AllowedMethod: {0}", string.Join(", ",
rule.AllowedMethods.ToArray()));
        Console.WriteLine("AllowedOrigins: {0}", string.Join(", ",
rule.AllowedOrigins.ToArray()));
        Console.WriteLine("AllowedHeaders: {0}", string.Join(", ",
rule.AllowedHeaders.ToArray()));
        Console.WriteLine("ExposeHeader: {0}", string.Join(", ",
rule.ExposeHeaders.ToArray()));
    }
}
}
```

Mit dem REST API

Um eine CORS Konfiguration für Ihren Bucket festzulegen, können Sie den AWS Management Console. Wenn Ihre Bewerbung dies erfordert, können Sie REST Anfragen auch direkt senden. In

den folgenden Abschnitten der Amazon Simple Storage Service API Reference werden die REST API Aktionen im Zusammenhang mit der CORS Konfiguration beschrieben:

- [PutBucketCors](#)
- [GetBucketCors](#)
- [DeleteBucketCors](#)
- [OPTIONSObjekt](#)

Testen CORS

Um Ihre CORS Konfiguration zu testen, kann mit dieser OPTIONS Methode eine CORS Preflight-Anfrage gesendet werden, sodass der Server antworten kann, ob das Senden der Anfrage akzeptabel ist. Wenn Amazon S3 eine Preflight-Anfrage erhält, bewertet S3 die CORS Konfiguration für den Bucket und verwendet die erste CORSRule Regel, die der eingehenden Anfrage entspricht, um eine ursprungsübergreifende Anfrage zu aktivieren. Für die Übereinstimmung mit einer Regel müssen die folgenden Bedingungen erfüllt sei:

- Der Origin Header in einer CORS Anfrage an Ihren Bucket muss mit den Ursprüngen im AllowedOrigins Element in Ihrer Konfiguration übereinstimmen. CORS
- Die HTTP Methoden, die Access-Control-Request-Method in einer CORS Anfrage an Ihren Bucket angegeben sind, müssen mit der oder den Methoden übereinstimmen, die in dem AllowedMethods Element in Ihrer CORS Konfiguration aufgeführt sind.
- Die im Header einer Preflight-Anfrage aufgeführten Access-Control-Request-Headers Header müssen mit den Headern im AllowedHeaders Element in Ihrer Konfiguration übereinstimmen. CORS

Das Folgende ist ein Beispiel für eine Konfiguration. CORS Informationen zum Erstellen einer CORS Konfiguration finden Sie unter [Konfiguration CORS](#). Weitere Beispiele für eine CORS Konfiguration finden Sie unter [Elemente einer CORS Konfiguration](#).

JSON

```
[
  {
    "AllowedHeaders": [
      "Authorization"
    ],
  },
]
```

```
    "AllowedMethods": [
      "GET",
      "PUT",
      "POST",
      "DELETE"
    ],
    "AllowedOrigins": [
      "http://www.example1.com"
    ],
    "ExposeHeaders": [
      "x-amz-meta-custom-header"
    ]
  }
]
```

Um die CORS Konfiguration zu testen, können Sie mit dem folgenden CURL Befehl einen OPTIONS Preflight-Check senden. CURL ist ein Befehlszeilentool, das für die Interaktion mit S3 verwendet werden kann. Weitere Informationen finden Sie unter [CURL](#)

```
curl -v -X OPTIONS \
-H "Origin: http://www.example1.com" \
-H "Access-Control-Request-Method: PUT" \
-H "Access-Control-Request-Headers: Authorization" \
-H "Access-Control-Expose-Headers: x-amz-meta-custom-header" \
"http://bucket_name.s3.amazonaws.com/object_prefix_name"
```

Im obigen Beispiel wird der `curl -v -x OPTIONS` Befehl verwendet, um eine Preflight-Anfrage an S3 zu senden, um abzufragen, ob S3 erlaubt ist, eine PUT Anfrage für ein Objekt vom Cross-Origin zu senden. `http://www.example1.com` Die Header `Access-Control-Request-Headers` und `Access-Control-Expose-Headers` sind optional.

- Als Antwort auf den `Access-Control-Request-Method` Header in der OPTIONS Preflight-Anfrage gibt Amazon S3 die Liste der zulässigen Methoden zurück, wenn die angeforderten Methoden übereinstimmen.
- Als Antwort auf den `Access-Control-Request-Headers` Header in der OPTIONS Preflight-Anfrage gibt Amazon S3 die Liste der zulässigen Header zurück, wenn die angeforderten Header übereinstimmen.

- Als Antwort auf den Access-Control-Expose-Headers Header in der OPTIONS Preflight-Anfrage gibt Amazon S3 eine Liste der zulässigen Header zurück, wenn die angeforderten Header mit den erlaubten Headern übereinstimmen, auf die von Skripten zugegriffen werden kann, die im Browser ausgeführt werden.

Note

Wenn beim Senden einer Preflight-Anfrage einer der CORS Anforderungsheader nicht zulässig ist, wird keiner der Antwortheader zurückgegeben. CORS

Als Antwort auf diese OPTIONS Preflight-Anfrage erhalten Sie eine Antwort. 200 OK Allgemeine Fehlercodes, die beim Testen aufgetreten sind, CORS und weitere Informationen zur Lösung CORS verwandter Probleme finden Sie unter [Problembehandlung CORS](#).

```
< HTTP/1.1 200 OK
< Date: Fri, 12 Jul 2024 00:23:51 GMT
< Access-Control-Allow-Origin: http://www.example1.com
< Access-Control-Allow-Methods: GET, PUT, POST, DELETE
< Access-Control-Allow-Headers: Authorization
< Access-Control-Expose-Headers: x-amz-meta-custom-header
< Access-Control-Allow-Credentials: true
< Vary: Origin, Access-Control-Request-Headers, Access-Control-Request-Method
< Server: AmazonS3
< Content-Length: 0
```

Fehlerbehebung für CORS

Die folgenden Themen können Ihnen helfen, einige häufig auftretende CORS Probleme im Zusammenhang mit S3 zu beheben.

Themen

- [403 Unerlaubter Fehler: CORS ist für diesen Bucket nicht aktiviert](#)
- [403 Unzulässiger Fehler: Diese CORS Anfrage ist nicht zulässig](#)
- [Header wurden als Antwort nicht gefunden CORS](#)
- [Überlegungen zu CORS S3-Proxy-Integrationen](#)

403 Unerlaubter Fehler: CORS ist für diesen Bucket nicht aktiviert

Der folgende 403 Forbidden Fehler tritt auf, wenn eine ursprungsübergreifende Anfrage an Amazon S3 gesendet, aber nicht in Ihrem S3-Bucket konfiguriert CORS ist.

Fehler: HTTP/1.1 403 Forbidden CORS Response: CORS ist für diesen Bucket nicht aktiviert.

Bei der CORS Konfiguration handelt es sich um ein Dokument oder eine Richtlinie mit Regeln, die die Ursprünge, denen Sie den Zugriff auf Ihren Bucket erlauben, die Operationen (HTTPMethoden), die Sie für jeden Ursprung unterstützen, und andere vorgangsspezifische Informationen festlegen. Erfahren [Sie, wie Sie mithilfe der Amazon S3-Konsole CORS auf S3 konfigurieren](#) AWS SDKs, und RESTAPI. Weitere Informationen CORS und Beispiele für eine CORS Konfiguration finden Sie unter [Elements of CORS](#).

403 Unzulässiger Fehler: Diese CORS Anfrage ist nicht zulässig

Der folgende 403 Forbidden Fehler wird angezeigt, wenn eine CORS Regel in Ihrer CORS Konfiguration nicht mit den Daten in Ihrer Anfrage übereinstimmt.

Fehler: HTTP/1.1 403 Verbotene CORS Antwort: Diese CORS Anfrage ist nicht zulässig.

Daher kann dieser 403 Forbidden Fehler aus mehreren Gründen auftreten:

- Herkunft ist nicht erlaubt.
- Methoden sind nicht erlaubt.
- Angeforderte Header sind nicht erlaubt.

Für jede Anfrage, die Amazon S3 empfängt, müssen Sie eine CORS Regel in Ihrer CORS Konfiguration haben, die den Daten in Ihrer Anfrage entspricht.

Origin ist nicht erlaubt

Der Origin Header in einer CORS Anfrage an Ihren Bucket muss mit den Ursprüngen im AllowedOrigins Element in Ihrer CORS Konfiguration übereinstimmen. Ein Platzhalterzeichen ("*") im AllowedOrigins Element würde allen HTTP Methoden entsprechen. Weitere Informationen zum Aktualisieren des AllowedOrigins Elements finden Sie unter [Konfiguration der gemeinsamen Nutzung von Ressourcen zwischen verschiedenen Quellen \(\) CORS](#).

Wenn beispielsweise nur die `http://www.example1.com` Domäne im `AllowedOrigins` Element enthalten ist, würde bei einer von der `http://www.example2.com` Domäne gesendeten CORS Anfrage der 403 Forbidden Fehler angezeigt.

Das folgende Beispiel zeigt einen Teil einer CORS Konfiguration, die die `http://www.example1.com` Domain in das `AllowedOrigins` Element einbezieht.

```
"AllowedOrigins":[
  "http://www.example1.com"
]
```

Damit eine von der `http://www.example2.com` Domain gesendete CORS Anfrage erfolgreich ist, muss die `http://www.example2.com` Domain in das `AllowedOrigins` CORS Konfigurationselement aufgenommen werden.

```
"AllowedOrigins":[
  "http://www.example1.com"
  "http://www.example2.com"
]
```

Methoden sind nicht erlaubt

Die HTTP Methoden, die `Access-Control-Request-Method` in einer CORS Anfrage an Ihren Bucket angegeben sind, müssen mit der oder den Methoden übereinstimmen, die in dem `AllowedMethods` Element in Ihrer CORS Konfiguration aufgeführt sind. Ein Platzhalterzeichen ("`*`") in `AllowedMethods` würde allen HTTP Methoden entsprechen. Weitere Informationen zum Aktualisieren des `AllowedOrigins` Elements finden Sie unter [Cross-Origin Resource Sharing \(\) CORS konfigurieren](#).

In einer CORS Konfiguration können Sie die folgenden Methoden im `AllowedMethods` Element angeben:

- GET
- PUT
- POST
- DELETE
- HEAD

Das folgende Beispiel zeigt einen Teil einer CORS Konfiguration, die die GET Methode im AllowedMethods Element enthält. Nur Anfragen, die die GET Methode enthalten, würden erfolgreich sein.

```
"AllowedMethods": [
  "GET"
]
```

Wenn eine HTTP Methode (z. B. PUT) in einer CORS Anfrage verwendet oder in einer CORS Preflight-Anfrage an Ihren Bucket enthalten war, die Methode jedoch nicht in Ihrer CORS Konfiguration vorhanden ist, würde die Anfrage zu einem 403 Forbidden Fehler führen. Um diese CORS Anfrage oder CORS Pre-Flight-Anfrage zuzulassen, muss die PUT Methode zu Ihrer CORS Konfiguration hinzugefügt werden.

```
"AllowedMethods": [
  "GET"
  "PUT"
]
```

Angeforderte Header sind nicht zulässig

Die im Header einer Pre-Flight-Anfrage aufgeführten Access-Control-Request-Headers Header müssen mit den Headern im AllowedHeaders Element in Ihrer Konfiguration übereinstimmen. CORS Eine Liste gängiger Header, die in Anfragen an Amazon S3 verwendet werden können, finden Sie unter [Allgemeine Anforderungsheader](#). Weitere Informationen zur Aktualisierung des AllowedHeaders Elements finden Sie unter [Konfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen](#) (). CORS

Das folgende Beispiel zeigt einen Teil einer CORS Konfiguration, die den Authorization Header im AllowedHeaders Element enthält. Nur Anfragen für den Authorization Header wären erfolgreich.

```
"AllowedHeaders": [
  "Authorization"
]
```

Wenn ein Header (zum Beispiel) in einer CORS Anfrage enthalten Content-MD5 war, der Header jedoch nicht in Ihrer CORS Konfiguration vorhanden ist, würde die Anfrage zu einem 403 Forbidden Fehler führen. Um diese CORS Anfrage zuzulassen, muss der Content-MD5 Header

zu Ihrer CORS Konfiguration hinzugefügt werden. Wenn Sie `Authorization` sowohl `Content-MD5` Header als auch `Header` in einer CORS Anfrage an Ihren Bucket übergeben möchten, stellen Sie sicher, dass beide Header im `AllowedHeaders` Element in Ihrer CORS Konfiguration enthalten sind.

```
"AllowedHeaders": [  
  "Authorization"  
  "Content-MD5"  
]
```

Header wurden als Antwort nicht gefunden CORS

Das `ExposeHeaders` Element in Ihrer CORS Konfiguration identifiziert, welche Antwortheader Sie Skripten und Anwendungen, die in Browsern ausgeführt werden, als Antwort auf eine CORS Anfrage zugänglich machen möchten.

Wenn Ihre in Ihrem S3-Bucket gespeicherten Objekte neben den Antwortdaten auch benutzerdefinierte Metadaten (z. B. `x-amz-meta-custom-header`) enthalten, könnte dieser benutzerdefinierte Header zusätzliche Metadaten oder Informationen enthalten, auf die Sie über Ihren JavaScript clientseitigen Code zugreifen möchten. Standardmäßig blockieren Browser jedoch aus Sicherheitsgründen den Zugriff auf benutzerdefinierte Header. Um Ihrem Client den Zugriff auf benutzerdefinierte Header JavaScript zu ermöglichen, müssen Sie den Header in Ihre Konfiguration aufnehmen. CORS

Im folgenden Beispiel ist der `x-amz-meta-custom-header1` Header im Element enthalten. `ExposeHeaders` Das `x-amz-meta-custom-header2` ist nicht im `ExposeHeaders` Element enthalten und fehlt in der CORS Konfiguration. In der Antwort würden nur die im `ExposeHeaders` Element enthaltenen Werte zurückgegeben. Wenn die Anfrage den `x-amz-meta-custom-header2` Header im `Access-Control-Expose-Headers` Header enthalten würde, würde die Antwort trotzdem `200 OK` zurückgeben. Es `x-amz-meta-custom-header` würde jedoch zum Beispiel nur der zulässige Header zurückgegeben und in der Antwort angezeigt.

```
"ExposeHeaders": [  
  "x-amz-meta-custom-header1"  
]
```

Um sicherzustellen, dass alle Header in der Antwort erscheinen, fügen Sie dem `ExposeHeaders` Element in Ihrer CORS Konfiguration alle zulässigen Header hinzu, wie unten gezeigt.

```
"ExposeHeaders": [
  "x-amz-meta-custom-header1",
  "x-amz-meta-custom-header2"
]
```

Überlegungen zu CORS S3-Proxy-Integrationen

Wenn bei Ihnen Fehler auftreten und Sie die CORS Konfiguration Ihres S3-Buckets bereits überprüft haben und die Cross-Origin-Anfrage an Proxys wie gesendet wird AWS CloudFront, versuchen Sie Folgendes:

- Konfigurieren Sie die Einstellungen, um die OPTIONS Methode für Anfragen zuzulassen. HTTP
- Konfigurieren Sie den Proxy so, dass er die folgenden Header weiterleitet: `OriginAccess-Control-Request-Headers`, und `Access-Control-Request-Method`.

Einige Proxys bieten vordefinierte Funktionen für Anfragen. CORS In können Sie beispielsweise eine Richtlinie konfigurieren CloudFront, die die Header enthält, die CORS Anfragen ermöglichen, wenn der Ursprung ein S3-Bucket ist. Weitere Informationen finden Sie unter [Steuern von Ursprungsanfragen mit einer Richtlinie](#) und [Verwenden von Richtlinien für verwaltete Ursprungsanfragen](#) im CloudFront Entwicklerhandbuch.

Entwickeln mit Amazon S3

Dieser Abschnitt enthält entwicklerbezogene Themen zur Verwendung von Amazon S3. Weitere Informationen finden Sie in den nachstehenden Themen.

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Senden von Anforderungen](#)
- [Entwickeln mit Amazon S3 über die AWS CLI](#)
- [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#)
- [Entwickeln mit Amazon S3 unter Verwendung der REST-API](#)
- [Behandlung von REST- und SOAP-Fehlern](#)
- [Referenz für Entwickler](#)

Senden von Anforderungen

Amazon S3 ist ein REST Service. Sie können Anfragen an Amazon S3 senden, indem Sie die REST API oder die AWS SDK (siehe [Beispielcode und Bibliotheken](#)) [Wrapper-Bibliotheken](#) verwenden RESTAPI, die das zugrunde liegende Amazon S3 umschließen und so Ihre Programmieraufgaben vereinfachen.

Jede Interaktion mit Amazon S3 erfolgt entweder authentifiziert oder anonym. Die Authentifizierung ist ein Vorgang, bei dem die Identität des Auftraggebers, der auf ein Amazon Web Services (AWS) Produkt zugreifen möchte, überprüft wird. Authentifizierte Anforderung müssen einen Signaturwert enthalten, der den Sender der Anforderung authentifiziert. Der Signaturwert wird teilweise aus den Zugriffsschlüsseln des Anforderers (AWS Zugriffsschlüssel-ID und geheimer Zugriffsschlüssel) generiert. Weitere Informationen zum Abrufen von Zugriffsschlüsseln finden Sie unter [Wie erhalte ich Sicherheitsanmeldeinformationen?](#) in der Allgemeine AWS-Referenz.

Wenn Sie den verwenden AWS SDK, berechnen die Bibliotheken die Signatur anhand der von Ihnen bereitgestellten Schlüssel. Wenn Sie jedoch direkte REST API Aufrufe in Ihrer Anwendung tätigen, müssen Sie den Code zur Berechnung der Signatur schreiben und der Anfrage hinzufügen.

Themen

- [Über Zugriffsschlüssel](#)
- [Anforderungsendpunkte](#)
- [Anfragen an Amazon S3 stellen über IPv6](#)
- [Senden von Anfragen unter Verwendung der AWS-SDKs](#)
- [Anfragen stellen mit dem REST API](#)

Über Zugriffsschlüssel

In den folgenden Abschnitten werden die Arten der Zugriffsschlüssel vorgestellt, die Sie verwenden können, um authentifizierte Anfragen zu senden.

AWS-Konto Zugriffstasten

Die Kontozugriffsschlüssel bieten vollen Zugriff auf die AWS Ressourcen, die dem Konto gehören. Im Folgenden finden Sie Beispiele für Zugriffsschlüssel:

- Zugriffsschlüssel-ID (eine alphanumerische Zeichenfolge mit 20 Zeichen). Zum Beispiel:
AKIAIOSFODNN7EXAMPLE
- Geheimer Zugriffsschlüssel (eine Zeichenfolge mit 40 Zeichen). Zum Beispiel: wJalrXUtnFEMI/
K7MDENG/bPxrFiCYEXAMPLEKEY

Die Zugriffsschlüssel-ID identifiziert eindeutig eine AWS-Konto. Sie können diese Zugriffsschlüssel verwenden, um authentifizierte Anforderungen an Amazon S3 zu senden.

IAMBenutzerzugriffsschlüssel

Sie können einen AWS-Konto für Ihr Unternehmen erstellen. Möglicherweise gibt es jedoch mehrere Mitarbeiter in der Organisation, die Zugriff auf die AWS Ressourcen Ihrer Organisation benötigen. Die gemeinsame Nutzung Ihrer AWS-Konto Zugangsschlüssel verringert die Sicherheit, und es ist möglicherweise nicht praktikabel, AWS-Konten für jeden Mitarbeiter individuelle Zugangsschlüssel zu erstellen. Darüber hinaus können Sie Ressourcen wie Buckets und Objekte nicht einfach

freigeben, da sie im Besitz verschiedener Konten sind. Um Ressourcen freizugeben, müssen Sie Berechtigungen gewähren. Dies bedeutet zusätzlichen Aufwand.

In solchen Szenarien können Sie AWS Identity and Access Management (IAM) verwenden, um Benutzer unter Ihnen AWS-Konto mit ihren eigenen Zugriffsschlüsseln zu erstellen und IAM Benutzerrichtlinien anzuhängen, die diesen Benutzern die entsprechenden Ressourcenzugriffsberechtigungen gewähren. IAM Damit Sie diese Benutzer besser verwalten können, können Sie Benutzergruppen erstellen und Berechtigungen auf Gruppenebene gewähren, die für alle Benutzer in dieser Gruppe gelten.

Diese Benutzer werden als IAM Benutzer bezeichnet, die Sie innerhalb von Ihnen erstellen und verwalten. AWS Das übergeordnete Konto steuert die Möglichkeiten der Benutzer, auf zuzugreifen AWS. Alle Ressourcen, die ein IAM Benutzer erstellt, unterliegen der Kontrolle des Elternteils und werden von diesem bezahlt AWS-Konto. Diese IAM Benutzer können authentifizierte Anfragen mit ihren eigenen Sicherheitsanmeldedaten an Amazon S3 senden. Weitere Informationen zum Erstellen und Verwalten von Benutzern unter Ihrem AWS-Konto Konto finden Sie auf der [Seite mit den AWS Identity and Access Management Produktdetails](#).

Temporäre Sicherheitsanmeldeinformationen

Neben der Erstellung von IAM Benutzern mit eigenen Zugriffsschlüsseln können Sie IAM auch jedem IAM Benutzer temporäre Sicherheitsanmeldedaten (temporäre Zugriffsschlüssel und ein Sicherheitstoken) gewähren, damit dieser auf Ihre AWS Dienste und Ressourcen zugreifen kann. Sie können Benutzer auch außerhalb von in Ihrem System verwaltete AWS. Diese Benutzer werden als verbundene Benutzer bezeichnet. Darüber hinaus können Benutzer Anwendungen sein, die Sie für den Zugriff auf Ihre AWS Ressourcen erstellen.

IAM bietet Ihnen die AWS Security Token Service API Möglichkeit, temporäre Sicherheitsanmeldedaten anzufordern. Sie können entweder die AWS STS API oder die verwenden AWS SDK, um diese Anmeldeinformationen anzufordern. Der API gibt temporäre Sicherheitsanmeldeinformationen (Zugriffsschlüssel-ID und geheimer Zugriffsschlüssel) und ein Sicherheitstoken zurück. Diese Anmeldeinformationen sind nur für die Dauer gültig, die Sie bei der Anforderung angeben. Sie verwenden die Zugriffsschlüssel-ID und den geheimen Schlüssel auf die gleiche Weise, wie Sie sie verwenden, wenn Sie Anfragen mit Ihren Zugangsschlüsseln AWS-Konto oder IAM Benutzerzugriffsschlüsseln senden. Zusätzlich muss jede Anfrage, die Sie an Amazon S3 senden, das Token enthalten.

Ein IAM Benutzer kann diese temporären Sicherheitsanmeldedaten für seinen eigenen Gebrauch anfordern oder sie an Verbundbenutzer oder -anwendungen weitergeben. Wenn Sie temporäre

Sicherheitsanmeldeinformationen für Verbundbenutzer anfordern, müssen Sie einen Benutzernamen und eine IAM Richtlinie angeben, in der die Berechtigungen definiert werden, die Sie diesen temporären Sicherheitsanmeldedaten zuordnen möchten. Der Verbundbenutzer kann nicht mehr Berechtigungen erhalten als der übergeordnete IAM Benutzer, der die temporären Anmeldeinformationen angefordert hat.

Sie können diese temporären Sicherheitsanmeldeinformationen beim Senden von Anfragen an Amazon S3 verwenden. Die API Bibliotheken berechnen den erforderlichen Signaturwert anhand dieser Anmeldeinformationen, um Ihre Anfrage zu authentifizieren. Wenn Sie beim Senden von Anfragen abgelaufene Anmeldeinformationen verwenden, lehnt Amazon S3 die Anfrage ab.

Informationen zum Signieren von Anfragen mithilfe temporärer Sicherheitsanmeldedaten in Ihren REST API Anfragen finden Sie unter [Anfragen signieren und authentifizieren REST](#). Hinweise zum Senden von Anfragen mit AWS SDKs finden Sie unter [Senden von Anfragen unter Verwendung der AWS-SDKs](#).

Weitere Informationen zur IAM Unterstützung temporärer Sicherheitsanmeldedaten finden Sie unter [Temporäre Sicherheitsanmeldedaten](#) im IAMBenutzerhandbuch.

Für zusätzliche Sicherheit können Sie beim Zugriff auf Ihre Amazon S3 S3-Ressourcen eine Multifaktor-Authentifizierung (MFA) verlangen, indem Sie eine Bucket-Richtlinie konfigurieren. Weitere Informationen finden Sie unter [Erforderlich MFA](#). Nachdem Sie MFA auf Ihre Amazon S3 S3-Ressourcen zugreifen müssen, können Sie auf diese Ressourcen nur zugreifen, indem Sie temporäre Anmeldeinformationen angeben, die mit einem MFA Schlüssel erstellt wurden. Weitere Informationen finden Sie auf der Detailseite zur [AWS Multi-Faktor-Authentifizierung](#) und [unter Konfiguration MFA — Geschützter API Zugriff](#) im IAMBenutzerhandbuch.

Anforderungsendpunkte

Sie senden REST Anfragen an den vordefinierten Endpunkt des Dienstes. Eine Liste aller AWS Dienste und der entsprechenden Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der [Allgemeine AWS-Referenz](#)

Anfragen an Amazon S3 stellen über IPv6

Amazon Simple Storage Service (Amazon S3) unterstützt zusätzlich zum Protokoll den Zugriff auf S3-Buckets mithilfe des IPv4 Internetprotokolls Version 6 (IPv6). Amazon S3 S3-Dual-Stack-Endpunkte unterstützen Anfragen an S3-Buckets über und. IPv6 IPv4 Für den Zugriff auf Amazon S3 fallen

keine zusätzlichen Gebühren an IPv6. Weitere Informationen zu Preisen finden Sie unter [Amazon-S3-Preise](#).

Themen

- [Erste Schritte für Anforderungen über IPv6](#)
- [Verwendung von IPv6-Adressen in IAM-Richtlinien](#)
- [Testen der IP-Adresskompatibilität](#)
- [Verwenden von Amazon-S3-Dual-Stack-Endpunkten](#)

Erste Schritte für Anforderungen über IPv6

Um eine Anfrage an einen S3-Bucket zu stellen IPv6, müssen Sie einen Dual-Stack-Endpunkt verwenden. Im nächsten Abschnitt wird beschrieben, wie Sie Anfragen mithilfe IPv6 von Dual-Stack-Endpunkten stellen.

Im Folgenden sind einige Dinge aufgeführt, die Sie wissen sollten, bevor Sie versuchen, auf einen Bucket zuzugreifen: IPv6

- Der Client und das Netzwerk, die auf den Bucket zugreifen, müssen für die Verwendung aktiviert sein IPv6.
- Sowohl Anfragen im virtuellen Hosting-Stil als auch im Pfadstil werden für IPv6 den Zugriff unterstützt. Weitere Informationen finden Sie unter [Amazon-S3-Dual-Stack-Endpunkte](#).
- Wenn Sie die Quell-IP-Adressfilterung in Ihren Benutzer- oder Bucket-Richtlinien AWS Identity and Access Management (IAM) verwenden, müssen Sie die Richtlinien aktualisieren, sodass sie IPv6 Adressbereiche einbeziehen. Weitere Informationen finden Sie unter [Verwendung von IPv6-Adressen in IAM-Richtlinien](#).
- Bei Verwendung IPv6 geben Serverzugriffs-Protokolldateien IP-Adressen in einem IPv6 Format aus. Sie müssen vorhandene Tools, Skripts und Software, die Sie zum Analysieren von Amazon S3 S3-Protokolldateien verwenden, aktualisieren, damit sie die IPv6 formatierten Remote IP Adressen analysieren können. Weitere Informationen finden Sie unter [Amazon-S3-Server-Zugriffsprotokollformat](#) und [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

Note

Wenn Sie Probleme im Zusammenhang mit dem Vorhandensein von IPv6 Adressen in Protokolldateien haben, wenden Sie sich an [AWS Support](#)

Anfragen mithilfe IPv6 von Dual-Stack-Endpunkten weiterleiten

Sie stellen Anfragen mit Amazon S3 API S3-Aufrufen über IPv6 Dual-Stack-Endpunkte. Die Amazon API S3-Operationen funktionieren auf die gleiche Weise, unabhängig davon, ob Sie mehr IPv6 oder mehr auf Amazon S3 zugreifen IPv4. Die Leistung sollte ebenfalls dieselbe bleiben.

Wenn Sie den verwenden RESTAPI, greifen Sie direkt auf einen Dual-Stack-Endpunkt zu. Weitere Informationen finden Sie unter [Dual-Stack-Endpunkte](#).

Wenn Sie AWS Command Line Interface (AWS CLI) und verwenden AWS SDKs, können Sie einen Parameter oder ein Flag verwenden, um zu einem Dual-Stack-Endpunkt zu wechseln. Sie können den Dual-Stack-Endpunkt auch direkt als Override des Amazon-S3-Endpunkts in der Konfigurationsdatei angeben.

Sie können einen Dual-Stack-Endpunkt verwenden, um über IPv6 eine der folgenden Optionen auf einen Bucket zuzugreifen:

- Das AWS CLI, siehe [Verwenden von Dual-Stack-Endpunkten aus dem AWS CLI](#).
- Die AWS SDKs, siehst du [Dual-Stack-Endpunkte von der AWS -SDKs verwenden](#).
- Die RESTAPI, siehst du [Anfragen an Dual-Stack-Endpunkte stellen mit dem REST API](#).

Funktionen, die über IPv6 nicht zur Verfügung stehen

Die folgende Funktion wird derzeit nicht unterstützt, wenn auf einen S3-Bucket zugegriffen wird IPv6: Statisches Website-Hosting von einem S3-Bucket aus.

Verwendung von IPv6-Adressen in IAM-Richtlinien

Bevor Sie versuchen IPv6, mithilfe eines Buckets auf einen Bucket zuzugreifen, müssen Sie sicherstellen, dass alle IAM Benutzer- oder S3-Bucket-Richtlinien, die für die IP-Adressfilterung verwendet werden, aktualisiert werden, sodass sie IPv6 Adressbereiche enthalten. IP-Adressfilterrichtlinien, die nicht für den Umgang mit IPv6 Adressen aktualisiert wurden, können dazu führen, dass Clients fälschlicherweise Zugriff auf den Bucket verlieren oder erhalten, wenn sie ihn verwenden IPv6. Weitere Informationen zur Verwaltung von Zugriffsberechtigungen mit IAM finden Sie unter [Identity and Access Management für Amazon S3](#).

IAM Richtlinien, die IP-Adressen filtern, verwenden [Bedingungsoperatoren für IP-Adressen](#). Die folgende Bucket-Richtlinie identifiziert den Bereich 54.240.143.* der zulässigen IPv4 Adressen

mithilfe von Operatoren für IP-Adressbedingungen. Alle IP-Adressen außerhalb dieses Bereichs erhalten keinen Zugriff auf den Bucket (`examplebucket`). Da sich alle IPv6 Adressen außerhalb des zulässigen Bereichs befinden, verhindert diese Richtlinie, dass IPv6 Adressen darauf zugreifen können. `examplebucket`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::examplebucket/*",
      "Condition": {
        "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

Sie können das Condition Element der Bucket-Richtlinie so ändern, dass sowohl IPv4 (`54.240.143.0/24`) als auch IPv6 (`2001:DB8:1234:5678::/64`) Adressbereiche zulässig sind, wie im folgenden Beispiel gezeigt. Sie können denselben Condition Blocktyp wie im Beispiel verwenden, um sowohl Ihre IAM Benutzer- als auch Ihre Bucket-Richtlinien zu aktualisieren.

```
"Condition": {
  "IpAddress": {
    "aws:SourceIp": [
      "54.240.143.0/24",
      "2001:DB8:1234:5678::/64"
    ]
  }
}
```

Vor der Verwendung müssen IPv6 Sie alle relevanten IAM Benutzer- und Bucket-Richtlinien aktualisieren, die IP-Adressfilterung verwenden. Es wird nicht empfohlen, die IP-Adressfilterung in Bucket-Richtlinien zu verwenden.

Sie können Ihre IAM Benutzerrichtlinien in der IAM Konsole unter <https://console.aws.amazon.com/iam/> überprüfen. Weitere Informationen zu IAM finden Sie im [IAMBenutzerhandbuch](#). Weitere

Informationen zum Bearbeiten von S3-Bucket-Richtlinien finden Sie unter [Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole](#).

Testen der IP-Adresskompatibilität

Wenn Sie Linux/Unix oder Mac OS X verwenden, können Sie testen, ob Sie auf einen Dual-Stack-Endpunkt zugreifen können, IPv6 indem Sie den `curl` Befehl verwenden, wie im folgenden Beispiel gezeigt:

Example

```
curl -v http://s3.dualstack.us-west-2.amazonaws.com/
```

Sie erhalten Informationen wie im folgenden Beispiel gezeigt zurück. Wenn Sie über IPv6 die verbundene IP-Adresse verbunden sind, wird dies eine Adresse sein. IPv6

```
* About to connect() to s3-us-west-2.amazonaws.com port 80 (#0)
* Trying IPv6 address... connected
* Connected to s3.dualstack.us-west-2.amazonaws.com (IPv6 address) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.18.1 (x86_64-unknown-linux-gnu) libcurl/7.18.1 OpenSSL/1.0.1t
zlib/1.2.3
> Host: s3.dualstack.us-west-2.amazonaws.com
```

Wenn Sie Microsoft Windows 7 oder Windows 10 verwenden, können Sie testen, ob Sie über IPv6 oder IPv4 mithilfe des `ping` Befehls auf einen Dual-Stack-Endpunkt zugreifen können, wie im folgenden Beispiel gezeigt.

```
ping ipv6.s3.dualstack.us-west-2.amazonaws.com
```

Verwenden von Amazon-S3-Dual-Stack-Endpunkten

Amazon-S3-Dual-Stack-Endpunkte unterstützen Anforderungen an S3-Buckets über IPv6 und IPv4. In diesem Abschnitt wird die Verwendung von Dual-Stack-Endpunkten beschrieben.

Themen

- [Amazon-S3-Dual-Stack-Endpunkte](#)

- [Verwenden von Dual-Stack-Endpunkten aus dem AWS CLI](#)
- [Dual-Stack-Endpunkte von der AWS -SDKs verwenden](#)
- [Verwenden von Dual-Stack-Endpunkte von der REST-API](#)

Amazon-S3-Dual-Stack-Endpunkte

Wenn Sie eine Anforderung an einen Dual-Stack-Endpunkt richten, wird die Bucket-URL in eine IPv6- oder eine IPv4-Adresse aufgelöst. Weitere Informationen zum Zugriff auf einen Bucket über IPv6 finden Sie unter [Anfragen an Amazon S3 stellen über IPv6](#).

Wenn Sie die REST-API verwenden, können Sie direkt auf einen Amazon-S3-Endpunkt zugreifen, indem Sie den Endpunktnamen (URI) verwenden. Über einen Dual-Stack-Endpunkt können Sie auf einen S3-Bucket zugreifen, indem Sie einen Virtual-Hosted-Style- oder Path-Style-Endpunktnamen verwenden. Amazon S3 unterstützt nur regionale Dual-Stack-Endpunktnamen, d. h. Sie müssen die Region als Teil des Namens angeben.

Verwenden Sie die folgenden Namenskonventionen für Endpunktnamen im Virtual-Hosted-Style und im Path-Style:

- Dual-Stack-Endpunkte im Virtual-Hosted-Style:

bucketname.s3.dualstack.aws-region.amazonaws.com

- Dual-Stack-Endpunkt im Path-Style:

s3.dualstack.aws-region.amazonaws.com/bucketname

Weitere Informationen zum Stil von Endpunktnamen finden Sie unter [Zugreifen auf einen Amazon-S3-Bucket und Auflisten des Buckets](#). Eine Liste der Amazon-S3-Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

Important

Für Dual-Stack-Endpunkte kann eine Transfer Acceleration verwendet werden. Weitere Informationen finden Sie unter [Erste Schritte mit Amazon S3 Transfer Acceleration](#).

Note

Die beiden Arten von VPC-Endpunkten für den Zugriff auf Amazon S3 (Schnittstellen-PC-Endpunkte und Gateway-VPC-Endpunkte) bieten keine Dual-Stack-Unterstützung. Weitere Informationen zu VPC-Endpunkten für Amazon S3 finden Sie unter [AWS PrivateLink für Amazon S3](#).

Wenn Sie die AWS SDKs AWS Command Line Interface (AWS CLI) und verwenden, können Sie einen Parameter oder ein Flag verwenden, um zu einem Dual-Stack-Endpunkt zu wechseln. Sie können den Dual-Stack-Endpunkt auch direkt als Override des Amazon-S3-Endpunkts in der Konfigurationsdatei angeben. In den folgenden Abschnitten wird beschrieben, wie Dual-Stack-Endpunkte aus den SDKs und den AWS CLI SDKs verwendet werden. AWS

Verwenden von Dual-Stack-Endpunkten aus dem AWS CLI

Dieser Abschnitt enthält Beispiele für AWS CLI Befehle, mit denen Anfragen an einen Dual-Stack-Endpunkt gestellt werden. Anweisungen zur Einrichtung von finden Sie AWS CLI unter [Entwickeln mit Amazon S3 über die AWS CLI](#).

Sie legen den Konfigurationswert `true` in einem Profil in Ihrer AWS Config Datei `use_dualstack_endpoint` auf fest, um alle Amazon S3 S3-Anfragen, die von den `s3api` AWS CLI Befehlen `s3` und gestellt werden, an den Dual-Stack-Endpunkt für die angegebene Region weiterzuleiten. Sie geben die Region in der Konfigurationsdatei oder in einem Befehl mit der Option `--region` an.

Bei der Verwendung von Dual-Stack-Endpunkten mit dem werden sowohl die AWS CLI `virtual` Adressierungsstile als auch beide `path` unterstützt. Der Adressierungsstil, der in der Konfigurationsdatei festgelegt wird, steuert, ob der Bucketname im Hostnamen enthalten oder Teil der URL ist. Standardmäßig versucht die CLI, den virtuellen Stil zu verwenden, wann immer das möglich ist, verwendet aber auch den Pfadstil, wenn das notwendig ist. Weitere Informationen finden Sie unter [AWS CLI -Amazon-S3-Konfiguration](#).

Sie können auch Konfigurationsänderungen über einen Befehl vornehmen, wie im folgenden Beispiel gezeigt, das im Standardprofil `use_dualstack_endpoint` auf `true` und `addressing_style` auf `virtual` setzt.

```
$ aws configure set default.s3.use_dualstack_endpoint true
$ aws configure set default.s3.addressing_style virtual
```

Wenn Sie einen Dual-Stack-Endpoint nur für bestimmte AWS CLI Befehle (nicht für alle Befehle) verwenden möchten, können Sie eine der folgenden Methoden verwenden:

- Sie können den Dual-Stack-Endpoint pro Befehl verwenden, indem Sie den Parameter `--endpoint-url` auf `https://s3.dualstack.aws-region.amazonaws.com` oder `http://s3.dualstack.aws-region.amazonaws.com` für jeden `s3-` oder `s3api-`Befehl setzen.

```
$ aws s3api list-objects --bucket bucketname --endpoint-url https://s3.dualstack.aws-region.amazonaws.com
```

- Sie können separate Profile in Ihrer AWS Config Datei einrichten. Legen Sie beispielsweise ein Profil an, das `use_dualstack_endpoint` auf `true` setzt, und ein Profil, das `use_dualstack_endpoint` nicht setzt. Wenn Sie einen Befehl ausführen, geben Sie an, welches Profil Sie verwenden wollen, abhängig davon, ob Sie den Dual-Stack-Endpoint verwenden wollen oder nicht.

Note

Wenn AWS CLI Sie die verwenden, können Sie derzeit keine Übertragungsbeschleunigung mit Dual-Stack-Endpoints verwenden. Die Unterstützung für AWS CLI wird jedoch bald verfügbar sein. Weitere Informationen finden Sie unter [Verwendung der AWS CLI](#).

Dual-Stack-Endpoints von der AWS -SDKs verwenden

Dieser Abschnitt enthält Beispiele für den Zugriff auf einen Dual-Stack-Endpoint mithilfe der SDKs.
AWS

AWS SDK for Java Beispiel eines Dual-Stack-Endpoints

Das folgende Beispiel veranschaulicht, wie Sie beim Erstellen eines Amazon-S3-Clients mit dem AWS SDK for Java Dual-Stack-Endpoints aktivieren.

Anweisungen zum Erstellen und Testen eines funktionierenden Java-Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;

public class DualStackEndpoints {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            // Create an Amazon S3 client with dual-stack endpoints enabled.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(clientRegion)
                .withDualstackEnabled(true)
                .build();

            s3Client.listObjects(bucketName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Wenn Sie das AWS SDK for Java unter Windows verwenden, müssen Sie möglicherweise die folgende Eigenschaft für die virtuelle Java-Maschine (JVM) festlegen:

```
java.net.preferIPv6Addresses=true
```

AWS Beispiel für einen Dual-Stack-Endpoint für .NET SDK

Wenn Sie das AWS SDK for .NET verwenden, verwenden Sie die `AmazonS3Config` Klasse, um die Verwendung eines Dual-Stack-Endpoints zu ermöglichen, wie im folgenden Beispiel gezeigt.

```
using Amazon;
```

```
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DualStackEndpointTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            var config = new AmazonS3Config
            {
                UseDualstackEndpoint = true,
                RegionEndpoint = bucketRegion
            };
            client = new AmazonS3Client(config);
            Console.WriteLine("Listing objects stored in a bucket");
            ListingObjectsAsync().Wait();
        }

        private static async Task ListingObjectsAsync()
        {
            try
            {
                var request = new ListObjectsV2Request
                {
                    BucketName = bucketName,
                    MaxKeys = 10
                };
                ListObjectsV2Response response;
                do
                {
                    response = await client.ListObjectsV2Async(request);

                    // Process the response.
                    foreach (S3Object entry in response.S3Objects)
                    {
                        Console.WriteLine("key = {0} size = {1}",
```

```
        entry.Key, entry.Size);
    }
    Console.WriteLine("Next Continuation Token: {0}",
response.NextContinuationToken);
    request.ContinuationToken = response.NextContinuationToken;
    } while (response.IsTruncated == true);
}
catch (AmazonS3Exception amazonS3Exception)
{
    Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
amazonS3Exception.ToString());
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.ToString());
}
}
}
```

Ein vollständiges .NET-Beispiel für die Auflistung von Objekten finden Sie unter [Programmgesteuertes Auflisten von Objektschlüsseln](#).

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Getting Started with the AWS SDK for .NET](#) im AWS SDK for .NET Developer Guide.

Verwenden von Dual-Stack-Endpunkte von der REST-API

Weitere Informationen über Anfragen an Dual-Stack-Endpunkte über die REST-API finden Sie unter [Anfragen an Dual-Stack-Endpunkte stellen mit dem REST API](#).

Senden von Anfragen unter Verwendung der AWS-SDKs

Themen

- [Anfragen stellen mit AWS-Konto oder IAM Benutzeranmeldedaten](#)
- [Anfragen mit temporären IAM Benutzeranmeldedaten stellen](#)
- [Anforderungen unter Verwendung temporärer Anmeldeinformationen verbundener Benutzer](#)

Sie können mit dem AWS-SDK oder durch direkte REST-API-Aufrufe innerhalb Ihrer Anwendung authentifizierte Anfragen an Amazon S3 senden. Die AWS-SDK-API verwendet die Anmeldeinformationen, die Sie für die Berechnung der Signatur für die Authentifizierung bereitstellen. Bei direkter Verwendung der REST-API in Ihren Anwendungen müssen Sie den benötigten Code zur Berechnung der Signatur für die Authentifizierung Ihrer Anforderung schreiben. Die Liste der verfügbaren AWS-SDKs finden Sie unter [Sample Code and Libraries \(Beispiel-Code und Bibliotheken\)](#).

Anfragen stellen mit AWS-Konto oder IAM Benutzeranmeldedaten

Sie können Ihre verwenden AWS-Konto oder Sicherheitsanmeldedaten von IAM Benutzern, um authentifizierte Anfragen an Amazon S3 zu senden. Dieser Abschnitt enthält Beispiele dafür, wie Sie authentifizierte Anfragen senden können, indem Sie AWS SDK for Java, AWS SDK for .NET, und AWS SDK for PHP. Für eine Liste der verfügbaren AWS SDKs, gehen Sie zu [Beispielcode und Bibliotheken](#).

Jeder von diesen AWS SDKs verwendet eine SDK spezifische Anbieterkette für Anmeldeinformationen, um Anmeldeinformationen zu finden und zu verwenden und Aktionen im Namen des Eigentümers der Anmeldeinformationen auszuführen. All diesen Anbieterketten für Anmeldeinformationen ist gemeinsam, dass sie alle nach Ihrem lokalen System suchen AWS Tools ebenfalls unterstützt.

Weitere Informationen finden Sie in den folgenden Themen:

Themen

- [Um ein lokales zu erstellen AWS Anmeldeinformationsdatei](#)
- [Senden authentifizierter Anfragen mit dem AWS SDKs](#)
- [Zugehörige Ressourcen](#)

Um ein lokales zu erstellen AWS Anmeldeinformationsdatei

Der einfachste Weg, Anmeldeinformationen für Ihr zu konfigurieren AWS SDKs ist die Verwendung eines AWS Tools ebenfalls unterstützt. Wenn Sie das AWS Command Line Interface (AWS CLI), vielleicht haben Sie bereits ein lokales AWS Anmeldeinformationsdatei konfiguriert. Andernfalls gehen Sie wie folgt vor, um eine Anmeldeinformationsdatei einzurichten:

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Erstellen Sie einen neuen Benutzer mit Berechtigungen, die auf die Services und Aktionen beschränkt sind, auf die Ihr Code Zugriff hat. Weitere Informationen zum Erstellen eines neuen Benutzers finden Sie unter [IAM Benutzer erstellen \(Konsole\)](#). Folgen Sie den Anweisungen in Schritt 8.
3. Wählen Sie „csv herunterladen“, um eine lokale Kopie Ihrer Datei zu speichern AWS Anmeldeinformationen.
4. Gehen Sie auf Ihrem Computer zu Ihrem Stammverzeichnis und erstellen Sie ein `.aws` Verzeichnis. Auf Unix-basierten Systemen (wie Linux oder OS X) befindet es sich an der folgenden Position:

```
~/ .aws
```

Unter Windows befindet es sich an der folgenden Position:

```
%HOMEPATH%\ .aws
```

5. Erstellen Sie in dem Verzeichnis `.aws` eine neue Datei namens `credentials`.
6. Öffnen Sie die `.csv` Datei mit den Anmeldeinformationen, die Sie von der IAM Konsole heruntergeladen haben, und kopieren Sie ihren Inhalt im folgenden Format in die `credentials` Datei:

```
[default]
aws_access_key_id = your_access_key_id
aws_secret_access_key = your_secret_access_key
```

7. Speichern Sie die `credentials`-Datei, und löschen Sie die `.csv`-Datei, die Sie in Schritt 3 heruntergeladen haben.

Ihre Datei mit den gemeinsam genutzten Anmeldeinformationen ist jetzt auf Ihrem lokalen Computer konfiguriert und kann mit dem verwendet werden AWS SDKs.

Senden authentifizierter Anfragen mit dem AWS SDKs

Verwenden Sie die AWS SDKsum authentifizierte Anfragen zu senden. Weitere Hinweise zum Senden authentifizierter Anfragen finden Sie unter [AWS Sicherheitsanmeldedaten](#) oder [IAMIdentity Center-Authentifizierung](#).

Java

Um authentifizierte Anfragen an Amazon S3 mit Ihrem zu senden AWS-Konto oder IAM Benutzeranmeldedaten, gehen Sie wie folgt vor:

- Erstellen Sie mit der `AmazonS3ClientBuilder`-Klasse eine `AmazonS3Client`-Instance.
- Führen Sie eine der `AmazonS3Client`-Methoden aus, um Anfragen an Amazon S3 zu senden. Der Client erstellt aus den von Ihnen angegebenen Anmeldeinformationen die erforderliche Signatur und nimmt sie in die Anforderung auf.

Das folgende Beispiel führt die vorhergehenden Aufgaben aus. Informationen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsRequest;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.s3.model.S3ObjectSummary;

import java.io.IOException;
import java.util.List;

public class MakingRequests {
```

```
public static void main(String[] args) throws IOException {
    Regions clientRegion = Regions.DEFAULT_REGION;
    String bucketName = "*** Bucket name ***";

    try {
        AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
            .withCredentials(new ProfileCredentialsProvider())
            .withRegion(clientRegion)
            .build();

        // Get a list of objects in the bucket, two at a time, and
        // print the name and size of each object.
        ListObjectsRequest listRequest = new
ListObjectsRequest().withBucketName(bucketName).withMaxKeys(2);
        ObjectListing objects = s3Client.listObjects(listRequest);
        while (true) {
            List<S3ObjectSummary> summaries = objects.getObjectSummaries();
            for (S3ObjectSummary summary : summaries) {
                System.out.printf("Object \"%s\" retrieved with size %d\n",
summary.getKey(), summary.getSize());
            }
            if (objects.isTruncated()) {
                objects = s3Client.listNextBatchOfObjects(objects);
            } else {
                break;
            }
        }
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Um authentifizierte Anfragen mit Ihrem zu senden AWS-Konto oder IAM Benutzeranmeldedaten:

- Erstellen Sie eine Instance der `AmazonS3Client`-Klasse.
- Führen Sie eine der `AmazonS3Client`-Methoden aus, um Anfragen an Amazon S3 zu senden. Der Client erstellt aus den von Ihnen angegebenen Anmeldeinformationen die erforderliche Signatur und nimmt sie in die Anforderung auf, die er an Amazon S3 sendet.

Weitere Informationen finden Sie unter [Anfragen stellen mit AWS-Konto oder IAM Benutzeranmeldedaten](#).

Note

- Sie können den `AmazonS3Client`-Client erstellen, ohne Sicherheitsanmeldeinformationen anzugeben. Anforderung, die unter Verwendung dieses Clients gesendet werden, sind anonym und haben keine Signatur. Amazon S3 gibt einen Fehler zurück, wenn Sie anonyme Anfragen für eine Ressource stellen, die nicht öffentlich verfügbar ist.
- Sie können ein AWS-Konto erstellen und die erforderlichen Benutzer erstellen. Sie können auch Anmeldeinformationen für diese Benutzer verwalten. Sie benötigen diese Anmeldeinformationen, um die Aufgabe im folgenden Beispiel auszuführen. Weitere Informationen finden Sie unter [Konfigurieren AWS Anmeldeinformationen](#) in der AWS SDK for .NET Leitfaden für Entwickler.

Sie können Ihre Anwendung dann auch so konfigurieren, dass sie aktiv Profile und Anmeldeinformationen abrufen und diese Anmeldeinformationen dann explizit bei der Erstellung eines AWS Service-Clients. Weitere Informationen finden Sie unter [Zugreifen auf Anmeldeinformationen und Profile in einer Anwendung](#) in der AWS SDK for .NET Entwicklerhandbuch.

Das folgende C#-Beispiel veranschaulicht, wie Sie die vorhergehenden Aufgaben ausführen. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
```

```
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class MakeS3RequestTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            using (client = new AmazonS3Client(bucketRegion))
            {
                Console.WriteLine("Listing objects stored in a bucket");
                ListingObjectsAsync().Wait();
            }
        }

        static async Task ListingObjectsAsync()
        {
            try
            {
                ListObjectsRequest request = new ListObjectsRequest
                {
                    BucketName = bucketName,
                    MaxKeys = 2
                };
                do
                {
                    ListObjectsResponse response = await
client.ListObjectsAsync(request);
                    // Process the response.
                    foreach (S3Object entry in response.S3Objects)
                    {
                        Console.WriteLine("key = {0} size = {1}",
                            entry.Key, entry.Size);
                    }

                    // If the response is truncated, set the marker to get the next
                    // set of keys.
                } while (response.IsTruncated);
            }
            catch { }
        }
    }
}
```

```
        if (response.IsTruncated)
        {
            request.Marker = response.NextMarker;
        }
        else
        {
            request = null;
        }
    } while (request != null);
}
catch (AmazonS3Exception e)
{
    Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
}
}
}
```

Funktionierende Beispiele finden Sie unter [Übersicht über Amazon-S3-Objekte](#) und [Bucket-Übersicht](#). Sie können diese Beispiele mit Ihrem testen AWS-Konto oder IAM Benutzeranmeldedaten.

Um beispielsweise alle Objektschlüssel in Ihrem Bucket aufzulisten, lesen Sie unter [Programmgesteuertes Auflisten von Objektschlüsseln](#).

PHP

In diesem Abschnitt wird erklärt, wie Sie eine Klasse aus Version 3 von verwenden AWS SDK for PHP um authentifizierte Anfragen mit Ihrem zu senden AWS-Konto oder IAM Benutzeranmeldedaten. Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Das folgende PHP Beispiel zeigt, wie der Client mithilfe Ihrer Sicherheitsanmeldedaten eine Anfrage stellt, um alle Buckets für Ihr Konto aufzulisten.

Example

```
require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => 'latest',
]);

// Retrieve the list of buckets.
$result = $s3->listBuckets();

try {
    // Retrieve a paginator for listing objects.
    $objects = $s3->getPaginator('ListObjects', [
        'Bucket' => $bucket
    ]);

    echo "Keys retrieved!" . PHP_EOL;

    // Print the list of objects to the page.
    foreach ($objects as $object) {
        echo $object['Key'] . PHP_EOL;
    }
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Note

Sie können den `S3Client`-Client erstellen, ohne Sicherheitsanmeldeinformationen anzugeben. Anforderung, die unter Verwendung dieses Clients gesendet werden, sind anonym und haben keine Signatur. Amazon S3 gibt einen Fehler zurück, wenn Sie anonyme Anfragen für eine Ressource stellen, die nicht öffentlich verfügbar ist.

Weitere Informationen finden Sie unter [Anonyme Clients erstellen](#) im [AWS SDK for PHP Dokumentation](#).

Funktionierende Beispiele finden Sie unter [Übersicht über Amazon-S3-Objekte](#). Sie können diese Beispiele mit Ihrem testen AWS-Konto oder IAM Benutzeranmeldedaten.

Ein Beispiel für die Auflistung der Objektschlüssel in einem Bucket finden Sie unter [Programmgesteuertes Auflisten von Objektschlüsseln](#).

Ruby

Bevor Sie Version 3 von verwenden können AWS SDK for Ruby Um Amazon S3 aufzurufen, müssen Sie die AWS Zugangsdaten, die SDK verwendet werden, um Ihren Zugriff auf Ihre Buckets und Objekte zu verifizieren. Wenn Sie gemeinsame Anmeldeinformationen in der eingerichtet haben AWS Das Anmeldeinformationsprofil auf Ihrem lokalen System kann Version 3 von SDK for Ruby diese Anmeldeinformationen verwenden, ohne dass Sie sie in Ihrem Code deklarieren müssen. Weitere Informationen zum Einrichten freigegebener Anmeldeinformationen finden Sie unter [Anfragen stellen mit AWS-Konto oder IAM Benutzeranmeldedaten](#).

Der folgende Ruby-Codeausschnitt verwendet die Anmeldeinformationen in einem gemeinsam genutzten AWS Anmeldeinformationsdatei auf einem lokalen Computer zur Authentifizierung einer Anfrage zum Abrufen aller Objektschlüsselnamen in einem bestimmten Bucket. Es führt die folgenden Aktionen aus:

1. Erstellt eine Instance der `Aws::S3::Client`-Klasse.
2. Sendet eine Anfrage an Amazon S3 durch Auflisten der Objekte in einem Bucket mit der `list_objects_v2`-Methode von `Aws::S3::Client`. Der Client generiert den erforderlichen Signaturwert aus den Anmeldeinformationen in AWS Anmeldeinformationsdatei auf Ihrem Computer und nimmt sie in die Anfrage auf, die an Amazon S3 gesendet wird.
3. Gibt das Array der Objektschlüsselnamen an das Terminal aus.

Example

```
# Prerequisites:  
# - An existing Amazon S3 bucket.  
  
require "aws-sdk-s3"
```

```
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if all operations succeed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_bucket_objects?(s3_client, 'amzn-s3-demo-bucket')
def list_bucket_objects?(s3_client, bucket_name)
  puts "Accessing the bucket named '#{bucket_name}'..."
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if objects.count.positive?
    puts "The object keys in this bucket are (first 50 objects):"
    objects.contents.each do |object|
      puts object.key
    end
  else
    puts "No objects found in this bucket."
  end

  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
  return false
end

# Example usage:
def run_me
  region = "us-west-2"
  bucket_name = "BUCKET_NAME"
  s3_client = Aws::S3::Client.new(region: region)

  exit 1 unless list_bucket_objects?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Wenn Sie kein lokales haben AWS Anmeldeinformationsdatei, Sie können die `Aws::S3::Client` Ressource weiterhin erstellen und Code für Amazon S3 S3-Buckets und

-Objekte ausführen. Anfragen, die mit Version 3 von SDK for Ruby gesendet werden, sind anonym und haben standardmäßig keine Signatur. Amazon S3 gibt einen Fehler zurück, wenn Sie anonyme Anfragen für eine Ressource stellen, die nicht öffentlich verfügbar ist.

Sie können den vorherigen Codeausschnitt SDK für Ruby-Anwendungen verwenden und erweitern, wie im folgenden, robusteren Beispiel gezeigt.

```
# Prerequisites:
# - An existing Amazon S3 bucket.

require "aws-sdk-s3"

# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if all operations succeed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_bucket_objects?(s3_client, 'amzn-s3-demo-bucket')
def list_bucket_objects?(s3_client, bucket_name)
  puts "Accessing the bucket named '#{bucket_name}'..."
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if objects.count.positive?
    puts "The object keys in this bucket are (first 50 objects):"
    objects.contents.each do |object|
      puts object.key
    end
  else
    puts "No objects found in this bucket."
  end

  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
  return false
end

# Example usage:
def run_me
```

```
    region = "us-west-2"
    bucket_name = "BUCKET_NAME"
    s3_client = Aws::S3::Client.new(region: region)

    exit 1 unless list_bucket_objects?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Go

Example

Das folgende Beispiel verwendet AWS Anmeldeinformationen, die von SDK for Go automatisch aus der Datei mit den gemeinsamen Anmeldeinformationen geladen werden.

```
package main

import (
    "context"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Storage Service
// (Amazon S3) client and list up to 10 buckets in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS
account?")
        fmt.Println(err)
        return
    }
    s3Client := s3.NewFromConfig(sdkConfig)
    count := 10
    fmt.Printf("Let's list up to %v buckets for your account.\n", count)
    result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
```

```
if err != nil {
    fmt.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
    return
}
if len(result.Buckets) == 0 {
    fmt.Println("You don't have any buckets!")
} else {
    if count > len(result.Buckets) {
        count = len(result.Buckets)
    }
    for _, bucket := range result.Buckets[:count] {
        fmt.Printf("\t%v\n", *bucket.Name)
    }
}
}
```

Zugehörige Ressourcen

- [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#)
- [AWS SDK for PHP für Amazon S3 Aws\ S3\ S3Client-Klasse](#)
- [AWS SDK for PHP Dokumentation](#)

Anfragen mit temporären IAM Benutzeranmeldedaten stellen

Importieren in &S3; AWS-Konto oder ein IAM Benutzer kann temporäre Sicherheitsanmeldedaten anfordern und diese verwenden, um authentifizierte Anfragen an Amazon S3 zu senden. Dieser Abschnitt enthält Beispiele für die Verwendung von AWS SDK for Java, .NET und PHP um temporäre Sicherheitsanmeldedaten zu erhalten und diese zur Authentifizierung Ihrer Anfragen an Amazon S3 zu verwenden.

Java

Ein IAM Benutzer oder ein AWS-Konto kann temporäre Sicherheitsanmeldedaten (siehe [Senden von Anforderungen](#)) anfordern, indem AWS SDK for Java und verwenden Sie sie, um auf Amazon S3 zuzugreifen. Diese Anmeldeinformationen laufen nach der angegebenen Sitzungsdauer ab.

Die Sitzungsdauer beträgt standardmäßig eine Stunde. Wenn Sie IAM Benutzeranmeldedaten verwenden, können Sie bei der Anforderung der temporären Sicherheitsanmeldedaten die Dauer von 15 Minuten bis zur maximalen Sitzungsdauer für die Rolle angeben. Weitere Informationen zu temporären Sicherheitsanmeldedaten finden Sie unter [Temporäre Sicherheitsanmeldedaten](#) im IAM Benutzerhandbuch. Weitere Informationen zum Senden von Anfragen finden Sie unter [Senden von Anforderungen](#).

So rufen Sie temporäre Sicherheitsanmeldeinformationen ab und greifen auf Amazon S3 zu

1. Erstellen Sie eine Instance der `AWSecurityTokenService`-Klasse. Weitere Informationen zum Bereitstellen von Anmeldeinformationen finden Sie unter [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).
2. Rufen Sie die temporären Sicherheitsanmeldedaten für die gewünschte Rolle ab, indem Sie die `assumeRole()` Methode des Security Token Service (STS) -Clients aufrufen.
3. Bündeln Sie die temporären Sicherheitsanmeldeinformationen in einem `BasicSessionCredentials`-Objekt. Sie verwenden dieses Objekt, um die temporären Sicherheitsanmeldeinformationen für Ihren Amazon-S3-Client bereitzustellen.
4. Erstellen Sie mit den temporären Sicherheitsanmeldeinformationen eine Instance der `AmazonS3Client`-Klasse. Mit diesem Client senden Sie Anfragen an Amazon S3. Wenn Sie beim Senden von Anfragen abgelaufene Anmeldeinformationen verwenden, gibt Amazon S3 einen Fehler zurück.

Note

Wenn Sie temporäre Sicherheitsanmeldedaten mit Ihrem AWS-Konto
Sicherheitsanmeldedaten: Die temporären Anmeldeinformationen sind nur für eine Stunde gültig. Sie können die Sitzungsdauer nur angeben, wenn Sie IAM Benutzeranmeldedaten verwenden, um eine Sitzung anzufordern.

Das folgende Beispiel listet einen Satz von Objektschlüsseln im angegebenen Bucket auf. Das Beispiel fordert temporäre Sicherheitsanmeldeinformationen für eine Sitzung an, die dann zum Senden einer authentifizierten Anfrage an Amazon S3 verwendet werden.

Wenn Sie das Beispiel mithilfe von IAM Benutzeranmeldedaten testen möchten, müssen Sie einen IAM Benutzer unter Ihrem Namen erstellen AWS-Konto. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie unter [Erstellen Ihrer ersten IAM Benutzer- und Administratorgruppe](#) im IAMBenutzerhandbuch.

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.AWSSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.securitytoken.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.AssumeRoleRequest;
import com.amazonaws.services.securitytoken.model.AssumeRoleResult;
import com.amazonaws.services.securitytoken.model.Credentials;

public class MakingRequestsWithIAMTempCredentials {
    public static void main(String[] args) {
        String clientRegion = "**** Client region ****";
        String roleARN = "**** ARN for role to be assumed ****";
        String roleSessionName = "**** Role session name ****";
        String bucketName = "**** Bucket name ****";
```

```
try {
    // Creating the STS client is part of your trusted code. It has
    // the security credentials you use to obtain temporary security
credentials.
    AWSSecurityTokenService stsClient =
AWSecurityTokenServiceClientBuilder.standard()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(clientRegion)
        .build();

    // Obtain credentials for the IAM role. Note that you cannot assume the
role of
    // an AWS root account;
    // Amazon S3 will deny access. You must use credentials for an IAM user
or an
    // IAM role.
    AssumeRoleRequest roleRequest = new AssumeRoleRequest()
        .withRoleArn(roleARN)
        .withRoleSessionName(roleSessionName);
    AssumeRoleResult roleResponse = stsClient.assumeRole(roleRequest);
    Credentials sessionCredentials = roleResponse.getCredentials();

    // Create a BasicSessionCredentials object that contains the credentials
you
    // just retrieved.
    BasicSessionCredentials awsCredentials = new BasicSessionCredentials(
        sessionCredentials.getAccessKeyId(),
        sessionCredentials.getSecretAccessKey(),
        sessionCredentials.getSessionToken());

    // Provide temporary security credentials so that the Amazon S3 client
    // can send authenticated requests to Amazon S3. You create the client
    // using the sessionCredentials object.
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .withCredentials(new
AWSStaticCredentialsProvider(awsCredentials))
        .withRegion(clientRegion)
        .build();

    // Verify that assuming the role worked and the permissions are set
correctly
    // by getting a set of object keys from the bucket.
    ObjectListing objects = s3Client.listObjects(bucketName);
```

```
        System.out.println("No. of Objects: " +
objects.getObjectSummaries().size());
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

.NET

Ein IAM Benutzer oder ein AWS-Konto kann temporäre Sicherheitsanmeldedaten anfordern mit AWS SDK for .NET und verwenden Sie sie, um auf Amazon S3 zuzugreifen. Diese Anmeldeinformationen laufen nach Ende der Sitzung ab.

Die Sitzungsdauer beträgt standardmäßig eine Stunde. Wenn Sie IAM Benutzeranmeldedaten verwenden, können Sie bei der Anforderung der temporären Sicherheitsanmeldedaten die Dauer von 15 Minuten bis zur maximalen Sitzungsdauer für die Rolle angeben. Weitere Informationen zu temporären Sicherheitsanmeldedaten finden Sie unter [Temporäre Sicherheitsanmeldedaten](#) im IAMBenutzerhandbuch. Weitere Informationen zum Senden von Anfragen finden Sie unter [Senden von Anforderungen](#).

So rufen Sie temporäre Sicherheitsanmeldeinformationen ab und greifen auf Amazon S3 zu

1. Erstellen Sie eine Instanz von AWS Security Token Service Kunde, `AmazonSecurityTokenServiceClient`. Weitere Informationen zum Bereitstellen von Anmeldeinformationen finden Sie unter [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#).
2. Starten Sie eine Sitzung, indem Sie die `GetSessionToken` Methode des STS Clients aufrufen, den Sie im vorherigen Schritt erstellt haben. Sie stellen für diese Methode Sitzungsdaten mithilfe eines `GetSessionTokenRequest`-Objekts bereit.

Die Methode gibt Ihre temporären Sicherheitsanmeldeinformationen zurück.

3. Bündeln Sie die temporären Sicherheitsanmeldeinformationen in einer Instance des Objekts `SessionAWSCredentials`. Sie verwenden dieses Objekt, um die temporären Sicherheitsanmeldeinformationen für Ihren Amazon-S3-Client bereitzustellen.
4. Erstellen Sie eine Instance der Klasse `AmazonS3Client`, indem Sie die temporären Sicherheitsanmeldeinformationen übergeben. Mit diesem Client senden Sie Anfragen an Amazon S3. Wenn Sie beim Senden der Anfragen abgelaufene Anmeldeinformationen verwenden, gibt Amazon S3 einen Fehler zurück.

Note

Wenn Sie temporäre Sicherheitsanmeldedaten mit Ihrem AWS-Konto Sicherheitsanmeldedaten, diese Anmeldeinformationen sind nur für eine Stunde gültig. Sie können eine Sitzungsdauer nur angeben, wenn Sie IAM Benutzeranmeldedaten verwenden, um eine Sitzung anzufordern.

Das folgende C#-Beispiel listet Objektschlüssel im angegebenen Bucket auf. Zur Veranschaulichung fordert das Beispiel temporäre Sicherheitsanmeldeinformationen für die standardmäßige, einstündige Sitzung an, die dann zum Senden einer authentifizierten Anfrage an Amazon S3 verwendet werden.

Wenn Sie das Beispiel mithilfe von IAM Benutzeranmeldedaten testen möchten, müssen Sie einen IAM Benutzer unter Ihrem Namen erstellen AWS-Konto. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie unter [Erstellen Ihrer ersten IAM Benutzer- und Administratorgruppe](#) im IAMBenutzerhandbuch. Weitere Informationen zum Senden von Anfragen finden Sie unter [Senden von Anforderungen](#).

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
```



```
namespace Amazon.DocSamples.S3
{
    class TempCredExplicitSessionStartTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            ListObjectsAsync().Wait();
        }

        private static async Task ListObjectsAsync()
        {
            try
            {
                // Credentials use the default AWS SDK for .NET credential search
chain.
                // On local development machines, this is your default profile.
                Console.WriteLine("Listing objects stored in a bucket");
                SessionAWSCredentials tempCredentials = await
GetTemporaryCredentialsAsync();

                // Create a client by providing temporary security credentials.
                using (s3Client = new AmazonS3Client(tempCredentials, bucketRegion))
                {
                    var listObjectRequest = new ListObjectsRequest
                    {
                        BucketName = bucketName
                    };
                    // Send request to Amazon S3.
                    ListObjectsResponse response = await
s3Client.ListObjectsAsync(listObjectRequest);
                    List<S3Object> objects = response.S3Objects;
                    Console.WriteLine("Object count = {0}", objects.Count);
                }
            }
            catch (AmazonS3Exception s3Exception)
            {
                Console.WriteLine(s3Exception.Message, s3Exception.InnerException);
            }
        }
    }
}
```

```
        catch (AmazonSecurityTokenServiceException stsException)
        {
            Console.WriteLine(stsException.Message,
stsException.InnerException);
        }
    }

    private static async Task<SessionAWSCredentials>
    GetTemporaryCredentialsAsync()
    {
        using (var stsClient = new AmazonSecurityTokenServiceClient())
        {
            var getSessionTokenRequest = new GetSessionTokenRequest
            {
                DurationSeconds = 7200 // seconds
            };

            GetSessionTokenResponse sessionTokenResponse =
                await
stsClient.GetSessionTokenAsync(getSessionTokenRequest);

            Credentials credentials = sessionTokenResponse.Credentials;

            var sessionCredentials =
                new SessionAWSCredentials(credentials.AccessKeyId,
                    credentials.SecretAccessKey,
                    credentials.SessionToken);

            return sessionCredentials;
        }
    }
}
```

PHP

Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Ein IAM Benutzer oder ein AWS-Konto kann temporäre Sicherheitsanmeldedaten mit Version 3 von anfordern AWS SDK for PHP. Es kann dann die temporären Anmeldeinformationen verwenden, um auf Amazon S3 zuzugreifen. Die Anmeldeinformationen laufen mit dem Ende der Sitzung ab.

Die Sitzungsdauer beträgt standardmäßig eine Stunde. Wenn Sie IAM Benutzeranmeldedaten verwenden, können Sie bei der Anforderung der temporären Sicherheitsanmeldedaten die Dauer von 15 Minuten bis zur maximalen Sitzungsdauer für die Rolle angeben. Weitere Informationen zu temporären Sicherheitsanmeldedaten finden Sie unter [Temporäre Sicherheitsanmeldedaten](#) im IAM Benutzerhandbuch. Weitere Informationen zum Senden von Anfragen finden Sie unter [Senden von Anforderungen](#).

Note

Wenn Sie temporäre Sicherheitsanmeldedaten mit Ihrem AWS-Konto Sicherheitsanmeldedaten, die temporären Sicherheitsanmeldedaten sind nur für eine Stunde gültig. Sie können die Sitzungsdauer nur angeben, wenn Sie IAM Benutzeranmeldedaten verwenden, um eine Sitzung anzufordern.

Example

Das folgende PHP Beispiel listet Objektschlüssel im angegebenen Bucket unter Verwendung temporärer Sicherheitsanmeldeinformationen auf. Das Beispiel fordert temporäre Sicherheitsanmeldeinformationen für eine standardmäßige, einstündige Sitzung an, die dann zum Senden einer authentifizierten Anfrage an Amazon S3 verwendet werden. Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Wenn Sie das Beispiel mithilfe von IAM Benutzeranmeldedaten testen möchten, müssen Sie einen IAM Benutzer unter Ihrem AWS-Konto. Informationen zum Erstellen eines IAM Benutzers finden Sie unter [Erstellen Ihrer ersten IAM Benutzer- und Administratorgruppe](#) im IAM Benutzerhandbuch. Beispiele für das Festlegen der Sitzungsdauer bei Verwendung von IAM Benutzeranmeldedaten zum Anfordern einer Sitzung finden Sie unter [Anfragen mit temporären IAM Benutzeranmeldedaten stellen](#).

```
require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
use Aws\Sts\StsClient;

$bucket = '*** Your Bucket Name ***';

$sts = new StsClient([
    'version' => 'latest',
```

```
'region' => 'us-east-1'
]);

$sessionToken = $sts->getSessionToken();

$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => 'latest',
    'credentials' => [
        'key' => $sessionToken['Credentials']['AccessKeyId'],
        'secret' => $sessionToken['Credentials']['SecretAccessKey'],
        'token' => $sessionToken['Credentials']['SessionToken']
    ]
]);

$result = $s3->listBuckets();

try {
    // Retrieve a paginator for listing objects.
    $objects = $s3->getPaginator('ListObjects', [
        'Bucket' => $bucket
    ]);

    echo "Keys retrieved!" . PHP_EOL;

    // List objects
    foreach ($objects as $object) {
        echo $object['Key'] . PHP_EOL;
    }
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

Ein IAM Benutzer oder ein AWS-Konto kann temporäre Sicherheitsanmeldedaten anfordern mit AWS SDK for Ruby und verwenden Sie sie, um auf Amazon S3 zuzugreifen. Diese Anmeldeinformationen laufen nach Ende der Sitzung ab.

Die Sitzungsdauer beträgt standardmäßig eine Stunde. Wenn Sie IAM Benutzeranmeldedaten verwenden, können Sie bei der Anforderung der temporären Sicherheitsanmeldedaten die Dauer von 15 Minuten bis zur maximalen Sitzungsdauer für die Rolle angeben. Weitere Informationen

zu temporären Sicherheitsanmeldedaten finden Sie unter [Temporäre Sicherheitsanmeldedaten](#) im IAMBenutzerhandbuch. Weitere Informationen zum Senden von Anfragen finden Sie unter [Senden von Anforderungen](#).

Note

Wenn Sie temporäre Sicherheitsanmeldedaten mit Ihrem AWS-Konto Sicherheitsanmeldedaten, die temporären Sicherheitsanmeldedaten sind nur für eine Stunde gültig. Sie können die Sitzungsdauer nur angeben, wenn Sie IAM Benutzeranmeldedaten verwenden, um eine Sitzung anzufordern.

Das folgende Ruby-Beispiel erstellt einen temporären Benutzer, damit dieser für eine Stunde die Elemente in einem bestimmten Bucket auflisten kann. Um dieses Beispiel verwenden zu können, benötigen Sie AWS Anmeldeinformationen, die über die erforderlichen Berechtigungen verfügen, um neue zu erstellen AWS Security Token Service (AWS STS) Clients und listet Amazon S3 S3-Buckets auf.

```
# Prerequisites:
# - A user in AWS Identity and Access Management (IAM). This user must
#   be able to assume the following IAM role. You must run this code example
#   within the context of this user.
# - An existing role in IAM that allows all of the Amazon S3 actions for all of the
#   resources in this code example. This role must also trust the preceding IAM
#   user.
# - An existing S3 bucket.

require "aws-sdk-core"
require "aws-sdk-s3"
require "aws-sdk-iam"

# Checks whether a user exists in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [Boolean] true if the user exists; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   exit 1 unless user_exists?(iam_client, 'my-user')
def user_exists?(iam_client, user_name)
```

```
    response = iam_client.get_user(user_name: user_name)
    return true if response.user.user_name
rescue Aws::IAM::Errors::NoSuchEntity
  # User doesn't exist.
rescue StandardError => e
  puts "Error while determining whether the user " \
    "'#{user_name}' exists: #{e.message}"
end

# Creates a user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [AWS:IAM::Types::User] The new user.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   user = create_user(iam_client, 'my-user')
#   exit 1 unless user.user_name
def create_user(iam_client, user_name)
  response = iam_client.create_user(user_name: user_name)
  return response.user
rescue StandardError => e
  puts "Error while creating the user '#{user_name}': #{e.message}"
end

# Gets a user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [AWS:IAM::Types::User] The existing user.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   user = get_user(iam_client, 'my-user')
#   exit 1 unless user.user_name
def get_user(iam_client, user_name)
  response = iam_client.get_user(user_name: user_name)
  return response.user
rescue StandardError => e
  puts "Error while getting the user '#{user_name}': #{e.message}"
end

# Checks whether a role exists in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
```

```
# @param role_name [String] The role's name.
# @return [Boolean] true if the role exists; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   exit 1 unless role_exists?(iam_client, 'my-role')
def role_exists?(iam_client, role_name)
  response = iam_client.get_role(role_name: role_name)
  return true if response.role.role_name
rescue StandardError => e
  puts "Error while determining whether the role " \
    "'#{role_name}' exists: #{e.message}"
end

# Gets credentials for a role in IAM.
#
# @param sts_client [Aws::STS::Client] An initialized AWS STS client.
# @param role_arn [String] The role's Amazon Resource Name (ARN).
# @param role_session_name [String] A name for this role's session.
# @param duration_seconds [Integer] The number of seconds this session is valid.
# @return [AWS::AssumeRoleCredentials] The credentials.
# @example
#   sts_client = Aws::STS::Client.new(region: 'us-west-2')
#   credentials = get_credentials(
#     sts_client,
#     'arn:aws:iam::123456789012:role/AmazonS3ReadOnly',
#     'ReadAmazonS3Bucket',
#     3600
#   )
#   exit 1 if credentials.nil?
def get_credentials(sts_client, role_arn, role_session_name, duration_seconds)
  Aws::AssumeRoleCredentials.new(
    client: sts_client,
    role_arn: role_arn,
    role_session_name: role_session_name,
    duration_seconds: duration_seconds
  )
rescue StandardError => e
  puts "Error while getting credentials: #{e.message}"
end

# Checks whether a bucket exists in Amazon S3.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
```

```
# @return [Boolean] true if the bucket exists; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless bucket_exists?(s3_client, 'amzn-s3-demo-bucket')
def bucket_exists?(s3_client, bucket_name)
  response = s3_client.list_buckets
  response.buckets.each do |bucket|
    return true if bucket.name == bucket_name
  end
rescue StandardError => e
  puts "Error while checking whether the bucket '#{bucket_name}' " \
    "exists: #{e.message}"
end

# Lists the keys and ETags for the objects in an Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the objects were listed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_objects_in_bucket?(s3_client, 'amzn-s3-demo-bucket')
def list_objects_in_bucket?(s3_client, bucket_name)
  puts "Accessing the contents of the bucket named '#{bucket_name}'..."
  response = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if response.count.positive?
    puts "Contents of the bucket named '#{bucket_name}' (first 50 objects):"
    puts "Name => ETag"
    response.contents.each do |obj|
      puts "#{obj.key} => #{obj.etag}"
    end
  else
    puts "No objects in the bucket named '#{bucket_name}'."
  end
  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
end
```


Zugehörige Ressourcen

- [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#)
- [AWS SDK for PHP für Amazon S3 Aws\ S3\ S3Client-Klasse](#)
- [AWS SDK for PHP Dokumentation](#)

Anforderungen unter Verwendung temporärer Anmeldeinformationen verbundener Benutzer

Sie können temporäre Sicherheitsanmeldedaten anfordern und sie Ihren Verbundbenutzern oder -anwendungen zur Verfügung stellen, die Zugriff auf Ihre AWS Ressourcen schätzen. Dieser Abschnitt enthält Beispiele dafür, wie Sie das verwenden können AWS SDKum temporäre Sicherheitsanmeldedaten für Ihre Verbundbenutzer oder Anwendungen zu erhalten und mit diesen Anmeldeinformationen authentifizierte Anfragen an Amazon S3 zu senden. Für eine Liste der verfügbaren AWS SDKs, siehe [Beispielcode und Bibliotheken](#).

Note

Sowohl die AWS-Konto und ein IAM Benutzer kann temporäre Sicherheitsanmeldeinformationen für Verbundbenutzer anfordern. Aus Sicherheitsgründen sollte jedoch nur ein IAM Benutzer mit den erforderlichen Berechtigungen diese temporären Anmeldeinformationen anfordern, um sicherzustellen, dass der Verbundbenutzer maximal die Berechtigungen des anfragenden IAM Benutzers erhält. In einigen Anwendungen kann es sinnvoll sein, einen IAM Benutzer mit bestimmten Berechtigungen zu erstellen, und zwar ausschließlich zu dem Zweck, Ihren Verbundbenutzern und -anwendungen temporäre Sicherheitsanmeldeinformationen zu gewähren.

Java

Sie können temporäre Sicherheitsanmeldedaten für Ihre Verbundbenutzer und -anwendungen bereitstellen, sodass diese authentifizierte Zugriffsanforderungen an Ihre Benutzer senden können AWS Ressourcen schätzen. Wenn Sie diese temporären Anmeldeinformationen anfordern, müssen Sie einen Benutzernamen und eine IAM Richtlinie angeben, die die Ressourcenberechtigungen beschreibt, die Sie gewähren möchten. Die Sitzungsdauer beträgt standardmäßig eine Stunde. Sie können explizit einen anderen Wert für die Dauer angeben, wenn Sie temporäre Sicherheitsanmeldeinformationen für verbundene Benutzer und Anwendungen anfordern.

Note

Für zusätzliche Sicherheit bei der Anforderung temporärer Sicherheitsanmeldeinformationen für Verbundbenutzer und -anwendungen empfehlen wir, einen dedizierten IAM Benutzer zu verwenden, der nur über die erforderlichen

Zugriffsberechtigungen verfügt. Der temporäre Benutzer, den Sie erstellen, kann nie mehr Berechtigungen erhalten als der IAM Benutzer, der die temporären Sicherheitsanmeldeinformationen angefordert hat. Weitere Informationen finden Sie unter [AWS Identity and Access Management FAQs](#).

Gehen Sie zum Bereitstellen von Sicherheitsanmeldeinformationen und zum Senden einer authentifizierten Anfrage für den Zugriff auf Ressourcen wie folgt vor:

- Erstellen Sie eine Instance der `AWSecurityTokenServiceClient`-Klasse.
- Starten Sie eine Sitzung, indem Sie die `getFederationToken()` Methode des Security Token Service (STS) -Clients aufrufen. Geben Sie Sitzungsinformationen, einschließlich des Benutzernamens und einer IAM Richtlinie, an, die Sie an die temporären Anmeldeinformationen anhängen möchten. Optional können Sie eine Sitzungsdauer angeben. Diese Methode gibt Ihre temporären Sicherheitsanmeldeinformationen zurück.
- Bündeln Sie die temporären Sicherheitsanmeldeinformationen in einer Instance des Objekts `BasicSessionCredentials`. Sie verwenden dieses Objekt, um die temporären Sicherheitsanmeldeinformationen für Ihren Amazon-S3-Client bereitzustellen.
- Erstellen Sie mit den temporären Sicherheitsanmeldeinformationen eine Instance der `AmazonS3Client`-Klasse. Mit diesem Client senden Sie Anfragen an Amazon S3. Wenn Sie beim Senden der Anfragen abgelaufene Anmeldeinformationen verwenden, gibt Amazon S3 einen Fehler zurück.

Example

Das folgende Beispiel listet die Schlüssel im angegebenen S3-Bucket auf. In dem Beispiel erhalten Sie temporäre Sicherheitsanmeldeinformationen für eine zweistündige Sitzung für Ihren verbundenen Benutzer und verwenden sie, um authentifizierte Anfragen an Amazon S3 zu senden. Um das Beispiel auszuführen, müssen Sie einen IAM Benutzer mit einer angehängten Richtlinie erstellen, die es dem Benutzer ermöglicht, temporäre Sicherheitsanmeldeinformationen anzufordern und Ihre AWS Ressourcen schätzen. Dies wird mit der folgenden Richtlinie erreicht:

```
{
  "Statement": [{
    "Action": ["s3:ListBucket",
      "sts:GetFederationToken*"],
  }],
}
```

```
        "Effect": "Allow",
        "Resource": "*"
    }
]
}
```

Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie unter [Creating Your First IAM User and Administrators Group](#) im IAMBenutzerhandbuch.

Nachdem Sie einen IAM Benutzer erstellt und die oben genannte Richtlinie angehängt haben, können Sie das folgende Beispiel ausführen. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter [Erste Schritte](#) im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.AWSSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.auth.policy.Policy;
import com.amazonaws.auth.policy.Resource;
import com.amazonaws.auth.policy.Statement;
import com.amazonaws.auth.policy.Statement.Effect;
import com.amazonaws.auth.policy.actions.S3Actions;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.securitytoken.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.Credentials;
import com.amazonaws.services.securitytoken.model.GetFederationTokenRequest;
import com.amazonaws.services.securitytoken.model.GetFederationTokenResult;

import java.io.IOException;

public class MakingRequestsWithFederatedTempCredentials {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Specify bucket name ***";
        String federatedUser = "*** Federated user name ***";
```

```
String resourceARN = "arn:aws:s3:::" + bucketName;

try {
    AWSSecurityTokenService stsClient = AWSSecurityTokenServiceClientBuilder
        .standard()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(clientRegion)
        .build();

    GetFederationTokenRequest getFederationTokenRequest = new
GetFederationTokenRequest();
    getFederationTokenRequest.setDurationSeconds(7200);
    getFederationTokenRequest.setName(federatedUser);

    // Define the policy and add it to the request.
    Policy policy = new Policy();
    policy.withStatements(new Statement(Effect.Allow)
        .withActions(S3Actions.ListObjects)
        .withResources(new Resource(resourceARN)));
    getFederationTokenRequest.setPolicy(policy.toJson());

    // Get the temporary security credentials.
    GetFederationTokenResult federationTokenResult =
stsClient.getFederationToken(getFederationTokenRequest);
    Credentials sessionCredentials = federationTokenResult.getCredentials();

    // Package the session credentials as a BasicSessionCredentials
    // object for an Amazon S3 client object to use.
    BasicSessionCredentials basicSessionCredentials = new
BasicSessionCredentials(
        sessionCredentials.getAccessKeyId(),
        sessionCredentials.getSecretAccessKey(),
        sessionCredentials.getSessionToken());
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .withCredentials(new
AWSStaticCredentialsProvider(basicSessionCredentials))
        .withRegion(clientRegion)
        .build();

    // To verify that the client works, send a listObjects request using
    // the temporary security credentials.
    ObjectListing objects = s3Client.listObjects(bucketName);
    System.out.println("No. of Objects = " +
objects.getObjectSummaries().size());
```

```
    } catch (AmazonServiceException e) {  
        // The call was transmitted successfully, but Amazon S3 couldn't process  
        // it, so it returned an error response.  
        e.printStackTrace();  
    } catch (SdkClientException e) {  
        // Amazon S3 couldn't be contacted for a response, or the client  
        // couldn't parse the response from Amazon S3.  
        e.printStackTrace();  
    }  
}  
}
```

.NET

Sie können temporäre Sicherheitsanmeldedaten für Ihre Verbundbenutzer und -anwendungen bereitstellen, damit diese authentifizierte Anfragen für den Zugriff auf Ihre AWS Ressourcen schätzen. Wenn Sie diese temporären Anmeldeinformationen anfordern, müssen Sie einen Benutzernamen und eine IAM Richtlinie angeben, die die Ressourcenberechtigungen beschreibt, die Sie gewähren möchten. Standardmäßig beträgt die Dauer einer Sitzung eine Stunde. Sie können explizit einen anderen Wert für die Dauer angeben, wenn Sie temporäre Sicherheitsanmeldeinformationen für verbundene Benutzer und Anwendungen anfordern. Informationen zum Senden authentifizierter Anfragen finden Sie unter [Senden von Anforderungen](#).

Note

Wenn Sie temporäre Sicherheitsanmeldeinformationen für Verbundbenutzer und -anwendungen anfordern, empfehlen wir aus Sicherheitsgründen, dass Sie einen dedizierten IAM Benutzer verwenden, der nur über die erforderlichen Zugriffsberechtigungen verfügt. Der temporäre Benutzer, den Sie erstellen, kann niemals mehr Berechtigungen erhalten als der IAM Benutzer, der die temporären Sicherheitsanmeldedaten angefordert hat. Weitere Informationen finden Sie unter [AWS Identity and Access Management FAQs](#).

Dazu führen Sie die folgenden Schritte aus:

- Erstellen Sie eine Instanz von AWS Security Token Service Kunde, `AmazonSecurityTokenServiceClient` Klasse.

- Starten Sie eine Sitzung, indem Sie die `GetFederationToken` Methode des STS Clients aufrufen. Sie müssen Sitzungsinformationen angeben, einschließlich des Benutzernamens und einer IAM Richtlinie, die Sie an die temporären Anmeldeinformationen anhängen möchten. Optional können Sie eine Sitzungsdauer angeben. Diese Methode gibt Ihre temporären Sicherheitsanmeldeinformationen zurück.
- Bündeln Sie die temporären Sicherheitsanmeldeinformationen in einer Instance des Objekts `SessionAWSCredentials`. Sie verwenden dieses Objekt, um die temporären Sicherheitsanmeldeinformationen für Ihren Amazon-S3-Client bereitzustellen.
- Erstellen Sie eine Instance der `AmazonS3Client`-Klasse, indem Sie die temporären Sicherheitsanmeldeinformationen übergeben. Sie verwenden diesen Client zum Senden von Anfragen an Amazon S3. Wenn Sie beim Senden der Anfragen abgelaufene Anmeldeinformationen verwenden, gibt Amazon S3 einen Fehler zurück.

Example

Das folgende C#-Beispiel listet die Schlüssel im angegebenen Bucket auf. In dem Beispiel erhalten Sie temporäre Sicherheitsanmeldeinformationen für eine zweistündige Sitzung für Ihren verbundenen Benutzer (User1) und verwenden sie, um authentifizierte Anfragen an Amazon S3 zu senden.

- Für diese Übung erstellen Sie einen IAM Benutzer mit minimalen Berechtigungen. Mit den Anmeldeinformationen dieses IAM Benutzers fordern Sie temporäre Anmeldeinformationen für andere Benutzer an. Dieses Beispiel listet nur die Objekte in einem bestimmten Bucket auf. Erstellen Sie einen IAM Benutzer mit der folgenden Richtlinie:

```
{
  "Statement": [{
    "Action": ["s3:ListBucket",
      "sts:GetFederationToken*"],
    "Effect": "Allow",
    "Resource": "*"
  }]
}
```

Die Richtlinie ermöglicht es dem IAM Benutzer, temporäre Sicherheitsanmeldedaten und Zugriffsberechtigungen nur für die Auflistung Ihrer AWS Ressourcen zu erhalten. Weitere

Informationen zum Erstellen eines IAM Benutzers finden Sie unter [Erstellen Ihrer IAM Benutzer- und Administratorgruppe](#) im IAMBenutzerhandbuch.

- Verwenden Sie die Sicherheitsanmeldedaten des IAM Benutzers, um das folgende Beispiel zu testen. Das Beispiel sendet unter Verwendung temporärer Sicherheitsanmeldeinformationen eine authentifizierte Anfrage an Amazon S3. Das Beispiel gibt die folgende Richtlinie an, wenn temporäre Sicherheitsanmeldeinformationen für den verbundenen Benutzer (User1) angefordert werden, womit der Zugriff auf eine Auflistung der Objekte in einem spezifischen Bucket eingeschränkt wird (YourBucketName). Sie müssen die Richtlinie aktualisieren, indem Sie Ihren eigenen bestehenden Bucket-Namen angeben.

```
{
  "Statement": [
    {
      "Sid": "1",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::YourBucketName"
    }
  ]
}
```

- Example

Aktualisieren Sie das folgende Beispiel und geben Sie den Bucket-Namen an, den Sie in der vorherigen Zugriffsrichtlinie für verbundene Benutzer angegeben haben. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Erste Schritte mit dem AWS SDK für .NET](#) in der AWS SDK für .NET Leitfaden für Entwickler.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TempFederatedCredentialsTest
```



```
{
    private const string bucketName = "*** bucket name ***";
    // Specify your bucket region (an example region is shown).
    private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
    private static IAmazonS3 client;

    public static void Main()
    {
        ListObjectsAsync().Wait();
    }

    private static async Task ListObjectsAsync()
    {
        try
        {
            Console.WriteLine("Listing objects stored in a bucket");
            // Credentials use the default AWS SDK for .NET credential search
chain.

            // On local development machines, this is your default profile.
            SessionAWSCredentials tempCredentials =
                await GetTemporaryFederatedCredentialsAsync();

            // Create a client by providing temporary security credentials.
            using (client = new AmazonS3Client(bucketRegion))
            {
                ListObjectsRequest listObjectRequest = new
ListObjectsRequest();
                listObjectRequest.BucketName = bucketName;

                ListObjectsResponse response = await
client.ListObjectsAsync(listObjectRequest);
                List<S3Object> objects = response.S3Objects;
                Console.WriteLine("Object count = {0}", objects.Count);

                Console.WriteLine("Press any key to continue...");
                Console.ReadKey();
            }
        }
        catch (AmazonS3Exception e)
        {
            Console.WriteLine("Error encountered ***. Message:'{0}' when
writing an object", e.Message);
        }
    }
}
```

```
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}'
when writing an object", e.Message);
        }
    }

    private static async Task<SessionAWSCredentials>
    GetTemporaryFederatedCredentialsAsync()
    {
        AmazonSecurityTokenServiceConfig config = new
    AmazonSecurityTokenServiceConfig();
        AmazonSecurityTokenServiceClient stsClient =
            new AmazonSecurityTokenServiceClient(
                config);

        GetFederationTokenRequest federationTokenRequest =
            new GetFederationTokenRequest();
        federationTokenRequest.DurationSeconds = 7200;
        federationTokenRequest.Name = "User1";
        federationTokenRequest.Policy = @"{
            ""Statement"":
            [
                {
                    ""Sid"":""Stmt1311212314284"",
                    ""Action"":[""s3:ListBucket""],
                    ""Effect"":""Allow"",
                    ""Resource"":""arn:aws:s3:::" + bucketName + @""
                }
            ]
        }
";

        GetFederationTokenResponse federationTokenResponse =
            await
    stsClient.GetFederationTokenAsync(federationTokenRequest);
        Credentials credentials = federationTokenResponse.Credentials;

        SessionAWSCredentials sessionCredentials =
            new SessionAWSCredentials(credentials.AccessKeyId,
                credentials.SecretAccessKey,
                credentials.SessionToken);

        return sessionCredentials;
    }
}
```

```
}  
}
```

PHP

In diesem Thema wird erklärt, wie Sie Klassen aus Version 3 von verwenden AWS SDK for PHP um temporäre Sicherheitsanmeldedaten für verbundene Benutzer und Anwendungen anzufordern und diese für den Zugriff auf in Amazon S3 gespeicherte Ressourcen zu verwenden. Für weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Sie können Ihren Verbundbenutzern und -anwendungen temporäre Sicherheitsanmeldedaten zur Verfügung stellen, damit diese authentifizierte Anfragen für den Zugriff auf Ihre Daten senden können AWS Ressourcen schätzen. Wenn Sie diese temporären Anmeldeinformationen anfordern, müssen Sie einen Benutzernamen und eine IAM Richtlinie angeben, die die Ressourcenberechtigungen beschreibt, die Sie gewähren möchten. Diese Anmeldeinformationen laufen mit dem Ende der Sitzung ab. Die Sitzungsdauer beträgt standardmäßig eine Stunde. Sie können explizit einen anderen Wert für die Dauer angeben, wenn Sie temporäre Sicherheitsanmeldeinformationen für verbundene Benutzer und Anwendungen anfordern. Weitere Informationen zu temporären Sicherheitsanmeldedaten finden Sie unter [Temporäre Sicherheitsanmeldedaten](#) im IAM Benutzerhandbuch. Informationen zum Bereitstellen temporärer Sicherheitsanmeldeinformationen für Ihre verbundenen Benutzer und Anwendungen finden Sie unter [Senden von Anforderungen](#).

Für zusätzliche Sicherheit bei der Anforderung temporärer Sicherheitsanmeldedaten für Verbundbenutzer und -anwendungen empfehlen wir, einen dedizierten IAM Benutzer zu verwenden, der nur über die erforderlichen Zugriffsberechtigungen verfügt. Der temporäre Benutzer, den Sie erstellen, kann nie mehr Berechtigungen erhalten als der IAM Benutzer, der die temporären Sicherheitsanmeldedaten angefordert hat. Hinweise zum Identitätsverbund finden Sie unter [AWS Identity and Access Management FAQs](#).

Weitere Informationen über AWS SDK für Ruby API gehen Sie zu [AWS SDK für Ruby - Version 2](#).

Example

Das folgende PHP Beispiel listet die Schlüssel im angegebenen Bucket auf. In dem Beispiel erhalten Sie temporäre Sicherheitsanmeldeinformationen für eine einstündige Sitzung für Ihren verbundenen Benutzer (User1). Sie verwenden die temporären Sicherheitsanmeldeinformationen dann zum Senden authentifizierter Anfragen an Amazon S3.

Für zusätzliche Sicherheit bei der Anforderung temporärer Anmeldeinformationen für andere verwenden Sie die Sicherheitsanmeldedaten eines IAM Benutzers, der berechtigt ist, temporäre Sicherheitsanmeldeinformationen anzufordern. Um sicherzustellen, dass der IAM Benutzer dem Verbundbenutzer nur die minimalen anwendungsspezifischen Berechtigungen gewährt, können Sie auch die Zugriffsberechtigungen dieses Benutzers einschränken. IAM Dieses Beispiel listet nur Objekte in einem bestimmten Bucket auf. Erstellen Sie einen IAM Benutzer mit der folgenden Richtlinie:

```
{
  "Statement": [{
    "Action": ["s3:ListBucket",
      "sts:GetFederationToken*"],
    "Effect": "Allow",
    "Resource": "*"
  }
]
```

Die Richtlinie ermöglicht es dem IAM Benutzer, temporäre Sicherheitsanmeldedaten und Zugriffsberechtigungen nur für die Auflistung Ihrer AWS Ressourcen schätzen. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie unter [Erstellen Ihrer ersten IAM Benutzer- und Administratorgruppe](#) im IAM Benutzerhandbuch.

Sie können jetzt die Sicherheitsanmeldedaten des IAM Benutzers verwenden, um das folgende Beispiel zu testen. Das Beispiel sendet unter Verwendung temporärer Sicherheitsanmeldeinformationen eine authentifizierte Anfrage an Amazon S3. Wenn temporäre Sicherheitsanmeldeinformationen für den verbundenen Benutzer (User1) angefordert werden, gibt das Beispiel die folgenden Richtlinie an, womit der Zugriff auf eine Auflistung der Objekte in einem bestimmten Bucket eingeschränkt wird. Aktualisieren Sie die Richtlinie mit Ihrem Bucket-Namen.

```
{
  "Statement": [
    {
      "Sid": "1",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::YourBucketName"
    }
  ]
}
```

```
}
```

Ersetzen Sie im folgenden Beispiel bei der Angabe der Richtlinien-Ressource `YourBucketName` durch den Namen Ihres Buckets:

```
require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
use Aws\Sts\StsClient;

$bucket = '*** Your Bucket Name ***';

// In real applications, the following code is part of your trusted code. It has
// the security credentials that you use to obtain temporary security credentials.
$sts = new StsClient([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Fetch the federated credentials.
$sessionToken = $sts->getFederationToken([
    'Name' => 'User1',
    'DurationSeconds' => '3600',
    'Policy' => json_encode([
        'Statement' => [
            'Sid' => 'randomstatementid' . time(),
            'Action' => ['s3:ListBucket'],
            'Effect' => 'Allow',
            'Resource' => 'arn:aws:s3:::' . $bucket
        ]
    ])
]);

// The following will be part of your less trusted code. You provide temporary
// security credentials so the code can send authenticated requests to Amazon S3.

$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => 'latest',
    'credentials' => [
        'key' => $sessionToken['Credentials']['AccessKeyId'],
        'secret' => $sessionToken['Credentials']['SecretAccessKey'],
```

```
        'token' => $sessionToken['Credentials']['SessionToken']
    ]
]);

try {
    $result = $s3->listObjects([
        'Bucket' => $bucket
    ]);
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

Sie können temporäre Sicherheitsanmeldedaten für Ihre Verbundbenutzer und -anwendungen bereitstellen, sodass diese authentifizierte Anfragen für den Zugriff auf Ihre Daten senden können AWS Ressourcen schätzen. Wenn Sie temporäre Anmeldeinformationen vom IAM Dienst anfordern, müssen Sie einen Benutzernamen und eine IAM Richtlinie angeben, die die Ressourcenberechtigungen beschreibt, die Sie gewähren möchten. Die Sitzungsdauer beträgt standardmäßig eine Stunde. Wenn Sie jedoch temporäre Anmeldeinformationen mithilfe von IAM Benutzeranmeldeinformationen anfordern, können Sie bei der Anforderung der temporären Sicherheitsanmeldeinformationen für Verbundbenutzer und -anwendungen explizit einen anderen Wert für die Dauer festlegen. Informationen zu temporären Sicherheitsanmeldeinformationen für Ihre verbundenen Benutzer und Anwendungen finden Sie unter [Senden von Anforderungen](#).

Note

Um die Sicherheit zu erhöhen, wenn Sie temporäre Sicherheitsanmeldeinformationen für Verbundbenutzer und -anwendungen anfordern, sollten Sie möglicherweise einen dedizierten IAM Benutzer verwenden, der nur über die erforderlichen Zugriffsberechtigungen verfügt. Der temporäre Benutzer, den Sie erstellen, kann nie mehr Berechtigungen erhalten als der IAM Benutzer, der die temporären Sicherheitsanmeldeinformationen angefordert hat. Weitere Informationen finden Sie unter [AWS Identity and Access Management FAQs](#).

Example

Das folgende Ruby-Codebeispiel gestattet es einem verbundenen Benutzer mit einem limitierten Satz von Berechtigungen Schlüssel im angegebenen Bucket aufzulisten.

```
# Prerequisites:
# - An existing Amazon S3 bucket.

require "aws-sdk-s3"
require "aws-sdk-iam"
require "json"

# Checks to see whether a user exists in IAM; otherwise,
# creates the user.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [Aws::IAM::Types::User] The existing or new user.
# @example
#   iam = Aws::IAM::Client.new(region: 'us-west-2')
#   user = get_user(iam, 'my-user')
#   exit 1 unless user.user_name
#   puts "User's name: #{user.user_name}"
def get_user(iam, user_name)
  puts "Checking for a user with the name '#{user_name}'..."
  response = iam.get_user(user_name: user_name)
  puts "A user with the name '#{user_name}' already exists."
  return response.user
# If the user doesn't exist, create them.
rescue Aws::IAM::Errors::NoSuchEntity
  puts "A user with the name '#{user_name}' doesn't exist. Creating this user..."
  response = iam.create_user(user_name: user_name)
  iam.wait_until(:user_exists, user_name: user_name)
  puts "Created user with the name '#{user_name}'."
  return response.user
rescue StandardError => e
  puts "Error while accessing or creating the user named '#{user_name}':
#{e.message}"
end

# Gets temporary AWS credentials for an IAM user with the specified permissions.
#
# @param sts [Aws::STS::Client] An initialized AWS STS client.
# @param duration_seconds [Integer] The number of seconds for valid credentials.
```

```

# @param user_name [String] The user's name.
# @param policy [Hash] The access policy.
# @return [Aws::STS::Types::Credentials] AWS credentials for API authentication.
# @example
#   sts = Aws::STS::Client.new(region: 'us-west-2')
#   credentials = get_temporary_credentials(sts, duration_seconds, user_name,
#     {
#       'Version' => '2012-10-17',
#       'Statement' => [
#         'Sid' => 'Stmt1',
#         'Effect' => 'Allow',
#         'Action' => 's3:ListBucket',
#         'Resource' => 'arn:aws:s3:::amzn-s3-demo-bucket'
#       ]
#     }
#   )
#   exit 1 unless credentials.access_key_id
#   puts "Access key ID: #{credentials.access_key_id}"
def get_temporary_credentials(sts, duration_seconds, user_name, policy)
  response = sts.get_federation_token(
    duration_seconds: duration_seconds,
    name: user_name,
    policy: policy.to_json
  )
  return response.credentials
rescue StandardError => e
  puts "Error while getting federation token: #{e.message}"
end

# Lists the keys and ETags for the objects in an Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the objects were listed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_objects_in_bucket?(s3_client, 'amzn-s3-demo-bucket')
def list_objects_in_bucket?(s3_client, bucket_name)
  puts "Accessing the contents of the bucket named '#{bucket_name}'..."
  response = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

```



```
if response.count.positive?
  puts "Contents of the bucket named '#{bucket_name}' (first 50 objects):"
  puts "Name => ETag"
  response.contents.each do |obj|
    puts "#{obj.key} => #{obj.etag}"
  end
else
  puts "No objects in the bucket named '#{bucket_name}'."
end
return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
end

# Example usage:
def run_me
  region = "us-west-2"
  user_name = "my-user"
  bucket_name = "amzn-s3-demo-bucket"

  iam = Aws::IAM::Client.new(region: region)
  user = get_user(iam, user_name)

  exit 1 unless user.user_name

  puts "User's name: #{user.user_name}"
  sts = Aws::STS::Client.new(region: region)
  credentials = get_temporary_credentials(sts, 3600, user_name,
    {
      "Version" => "2012-10-17",
      "Statement" => [
        "Sid" => "Stmt1",
        "Effect" => "Allow",
        "Action" => "s3:ListBucket",
        "Resource" => "arn:aws:s3:::#{bucket_name}"
      ]
    }
  )

  exit 1 unless credentials.access_key_id

  puts "Access key ID: #{credentials.access_key_id}"
  s3_client = Aws::S3::Client.new(region: region, credentials: credentials)
```

```
    exit 1 unless list_objects_in_bucket?(s3_client, bucket_name)
  end

  run_me if $PROGRAM_NAME == __FILE__
```

Zugehörige Ressourcen

- [Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs](#)
- [AWS SDK for PHP für Amazon S3 Aws\ S3\ S3Client-Klasse](#)
- [AWS SDK for PHP Dokumentation](#)

Anfragen stellen mit dem REST API

Dieser Abschnitt enthält Informationen darüber, wie Sie Anfragen an Amazon S3 S3-Endpunkte mithilfe von stellen können. REST API Eine Liste der Amazon-S3-Endpunkte finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

Konstruieren von S3-Hostnamen für Anfragen REST API

Amazon-S3-Endpunkte folgen der folgenden Struktur:

```
s3.Region.amazonaws.com
```

Amazon-S3-Zugriffspunkte-Endpunkte und Dual-Stack-Endpunkte folgen ebenfalls der Standardstruktur:

- Amazon-S3-Zugriffspunkte -s3-accesspoint.*Region*.amazonaws.com
- Dual-stack - s3.dualstack.*Region*.amazonaws.com

Eine vollständige Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter [Endpunkte und Kontingente von Amazon S3](#) in der Allgemeine Amazon Web Services-Referenz.

Virtuell gehostete und Pfad-Stil-Anforderungen

Wenn Sie Anfragen mit dem stellen RESTAPI, können Sie URIs für die Amazon S3 S3-Endpunkte den virtuellen Hosted-Stil oder den Pfad-Stil verwenden. Weitere Informationen finden Sie unter [Virtuelles Hosting bei Buckets](#).

Example Anforderung im virtuell gehosteten Stil

Nachfolgend finden Sie ein Beispiel für eine virtuell gehostete Anforderung zum Löschen der Datei `puppy.jpg` aus dem Bucket `examplebucket` in der Region USA West (Oregon). Weitere Informationen zum Wiederholen von Anforderungen finden Sie unter [Anforderungen im virtuellen Hosting-Format](#).

```
DELETE /puppy.jpg HTTP/1.1
Host: examplebucket.s3.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Example Pfadanforderung

Im Folgenden finden Sie ein Beispiel für eine Path-Style-Version derselben Anfrage.

```
DELETE /examplebucket/puppy.jpg HTTP/1.1
Host: s3.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Derzeit unterstützt Amazon S3 sowohl virtuell gehosteten Zugriff als auch Pfadzugriff. URL AWS-Regionen Path-Style URLs wird jedoch in future eingestellt. Weitere Informationen finden Sie im folgenden wichtigen Hinweis.

Weitere Informationen zu Anforderungen im Pfadformat finden Sie unter [Anforderungen im Pfadformat](#).

Important

Update (23. September 2020) — Um sicherzustellen, dass Kunden die Zeit haben, die sie für die Umstellung auf virtuell gehostete Systeme benötigen, haben wir beschlossen URLs, die Einstellung von Path-Style zu verzögern. URLs Weitere Informationen finden Sie unter [Amazon S3 Path Deprecation Plan – The Rest of the Story](#) im AWS News Blog.

Anfragen an Dual-Stack-Endpunkte stellen mit dem REST API

Wenn Sie den verwenden RESTAPI, können Sie direkt auf einen Dual-Stack-Endpunkt zugreifen, indem Sie einen Endpunktnamen im virtuellen Hosted-Stil oder einen Pfadstil verwenden (). URI Alle Amazon-S3-Dual-Stack-Endpunktnamen enthalten die Region. Im Gegensatz zu den reinen Standardendpunkten IPv4 verwenden sowohl virtuelle Host-Endpoints als auch Pfadendpunkte regionsspezifische Endpunktnamen.

Example Dual-Stack-Endpunkt-Anforderung im virtuell gehosteten Stil

Sie können in Ihrer REST Anfrage einen Endpunkt im Stil eines virtuellen Hosts verwenden, wie im folgenden Beispiel gezeigt, mit dem das Objekt aus dem Bucket abgerufen wird, der in der Region USA West (Oregon) benannt ist. puppy.jpg examplebucket

```
GET /puppy.jpg HTTP/1.1
Host: examplebucket.s3.dualstack.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Example Pfadanforderungen für Dual-Stack-Endpunkte

Sie können auch wie im folgenden Beispiel gezeigt einen Path-Style-Endpunkt in Ihrer Anfrage verwenden.

```
GET /examplebucket/puppy.jpg HTTP/1.1
Host: s3.dualstack.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Weitere Informationen zu Dual-Stack-Endpunkten finden Sie unter [Verwenden von Amazon-S3-Dual-Stack-Endpunkten](#).

Weitere Informationen zum Stellen von Anfragen mithilfe von finden Sie in den REST API folgenden Themen.

Themen

- [Virtuelles Hosting bei Buckets](#)

- [Anforderungsumleitung und die REST-API](#)

Virtuelles Hosting bei Buckets

Das virtuelle Hosting ist das beste Verfahren, um mehrere Websites mit einem Webserver zu unterstützen. Eine Möglichkeit, Websites in Ihren Amazon S3 REST API S3-Anfragen zu unterscheiden, besteht darin, den offensichtlichen Hostnamen der Anfrage zu verwenden — URI statt nur den Pfadnamen, der URI Teil von ist. Eine normale Amazon S3 REST S3-Anfrage spezifiziert einen Bucket unter Verwendung der ersten durch Schrägstrich getrennten Komponente des Request-Pfads. URI Stattdessen können Sie das virtuelle Amazon S3 S3-Hosting verwenden, um einen Bucket in einem REST API Aufruf mithilfe des HTTP Host Headers zu adressieren. In der Praxis interpretiert Amazon S3 Host so, dass die meisten Buckets automatisch (für begrenzte Anfragetypen) unter `https://bucket-name.s3.region-code.amazonaws.com` zur Verfügung stehen. Eine vollständige Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter [Amazon-S3-Endpunkte und -Kontingente](#) in der Allgemeine Amazon Web Services-Referenz.

Virtuelles Hosting hat auch andere Vorteile. Indem Sie Ihren Bucket nach Ihrem registrierten Domainnamen benennen und diesen Namen zu einem DNS Alias für Amazon S3 machen, können Sie Ihre Amazon S3 S3-Ressourcen vollständig anpassen, `http://my.bucket-name.com/` z. B. URL Sie können auch im „Stammverzeichnis“ des virtuellen Servers Ihres Buckets veröffentlichen. Diese Fähigkeit kann wichtig sein, da viele vorhandene Anwendungen nach Dateien an diesem Standardspeicherort suchen. Beispielsweise wird erwartet, dass `favicon.ico`, `robots.txt` und `crossdomain.xml` im Stamm gefunden werden können.

Important

Wenn Sie Buckets im Virtual-Hosting-Stil mit verwenden, stimmt das SSL Platzhalterzertifikat nur mit Buckets überein, die keine Punkte (.) enthalten. Um diese Einschränkung zu umgehen, verwenden oder schreiben Sie Ihre eigene Logik zur Zertifikatsverifizierung. HTTP Weitere Informationen finden Sie unter [Amazon S3 Path Deprecation Plan](#) im AWS News Blog.

Themen

- [Anforderungen im Pfadformat](#)
- [Anforderungen im virtuellen Hosting-Format](#)
- [HTTPHostHeader-Bucket-Spezifikation](#)

- [Beispiele](#)
- [Amazon S3 URLs mit CNAME Datensätzen anpassen](#)
- [Verknüpfen eines Host-Namens mit einem Amazon-S3-Bucket](#)
- [Einschränkungen](#)
- [Abwärtskompatibilität](#)

Anforderungen im Pfadformat

Derzeit unterstützt Amazon S3 sowohl virtuell gehosteten Zugriff als auch Pfadzugriff. URL AWS-Regionen Path-Style URLs wird jedoch in future eingestellt. Weitere Informationen finden Sie im folgenden wichtigen Hinweis.

In Amazon S3 URLs verwendet path-style das folgende Format:

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

Wenn Sie beispielsweise einen Bucket mit dem Namen amzn-s3-demo-bucket1 USA West (Oregon) erstellen und auf das puppy.jpg Objekt in diesem Bucket zugreifen möchten, können Sie den folgenden Pfadstil verwenden: URL

```
https://s3.us-west-2.amazonaws.com/amzn-s3-demo-bucket1/puppy.jpg
```

Important

Update (23. September 2020) — Um sicherzustellen, dass Kunden die Zeit haben, die sie für die Umstellung auf virtuell gehostete Systeme benötigen, haben wir beschlossen URLs, die die Einstellung von Path-Style zu verzögern. URLs Weitere Informationen finden Sie unter [Amazon S3 Path Deprecation Plan – The Rest of the Story](#) im AWS News Blog.

Warning

Vermeiden Sie beim Hosten von Website-Inhalten, auf die über einen Webbrowser zugegriffen wird, die Verwendung des Pfadformats, da dies das Sicherheitsmodell des Browsers „Selbe Origin“ beeinträchtigen URLs könnte. Zum Hosten von Website-Inhalten empfehlen wir, entweder S3-Website-Endpunkte oder eine Distribution zu verwenden. CloudFront Weitere Informationen finden Sie unter [Website-Endpunkte](#) und [Bereitstellen](#)

einer React-basierten Einzelseitenanwendung für Amazon S3 und CloudFront in den AWS Perspective Guidance Patterns.

Anforderungen im virtuellen Hosting-Format

Bei einem virtuell gehosteten System ist der Bucket-Name URI Teil des Domainnamens in der URL

Verwenden Sie im virtuellen Hosted-Stil von Amazon S3 das folgende Format: URLs

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

In diesem Beispiel ist `amzn-s3-demo-bucket1` der Bucket-Name, "US West (Oregon) (USA West (Oregon))" die Region und `puppy.png` der Schlüsselname:

```
https://amzn-s3-demo-bucket1.s3.us-west-2.amazonaws.com/puppy.png
```

HTTP**Host**Header-Bucket-Spezifikation

Solange Ihre GET Anfrage den SSL Endpunkt nicht verwendet, können Sie den Bucket für die Anfrage mithilfe des HTTP Host Headers angeben. Der Host Header in einer REST Anfrage wird wie folgt interpretiert:

- Wenn der Host Header weggelassen wird oder sein Wert `s3.region-code.amazonaws.com` ist, ist der Bucket für die Anfrage die erste durch Schrägstriche getrennte Komponente der Anfrage-URI und der Schlüssel für die Anfrage ist der Rest der Anfrage-URI. Dies ist die übliche Methode wie im ersten und zweiten Beispiel in diesem Abschnitt dargestellt. Das Weglassen des Host Headers ist nur für 1.0-Anfragen gültig. HTTP
- Wenn andernfalls der Wert des Host-Headers mit `.s3.region-code.amazonaws.com` endet, ist der Bucket-Name die führende Komponente im Wert des Host-Headers bis zu `.s3.region-code.amazonaws.com`. Der Schlüssel für die Anfrage ist Request-URI. Diese Interpretation stellt Buckets als Unterdomänen von `.s3.region-code.amazonaws.com` bereit (vgl. das dritte und vierte Beispiel in diesem Abschnitt).
- Andernfalls ist der Bucket für die Anfrage der Kleinbuchstabenwert des Host Headers, und der Schlüssel für die Anfrage ist der Request-URI. Diese Interpretation ist nützlich, wenn Sie denselben DNS Namen wie Ihren Bucket-Namen registriert und diesen Namen als kanonischen Namen (CNAME) -Alias für Amazon S3 konfiguriert haben. Das Verfahren zur Registrierung von Domainnamen und zur Konfiguration von CNAME DNS Datensätzen würde den Rahmen dieses

Handbuchs sprengen. Das Ergebnis wird jedoch anhand des letzten Beispiels in diesem Abschnitt veranschaulicht.

Beispiele

Dieser Abschnitt enthält Beispiele URLs und Anfragen.

Example — Pfadstil URLs und Anfragen

In diesem Beispiel wird Folgendes verwendet:

- Bucket-Name – `example.com`
- Region: USA Ost (Nord-Virginia)
- Schlüsselname – `homepage.html`

Das URL ist wie folgt:

```
http://s3.us-east-1.amazonaws.com/example.com/homepage.html
```

die Anforderung lautet wie folgt:

```
GET /example.com/homepage.html HTTP/1.1  
Host: s3.us-east-1.amazonaws.com
```

Die Anfrage mit HTTP 1.0 und dem Weglassen des Host Headers lautet wie folgt:

```
GET /example.com/homepage.html HTTP/1.0
```

[Informationen zu DNS -kompatiblen Namen finden Sie unter Einschränkungen.](#) Weitere Informationen zu Schlüsseln finden Sie unter [Schlüssel](#).

Example — URLs Virtual-Hosted — Stil und Anfragen

In diesem Beispiel wird Folgendes verwendet:

- Bucket-Name – `amzn-s3-demo-bucket1`
- Region - Europa (Irland)
- Schlüsselname – `homepage.html`

Das ist wie folgt: URL

```
http://amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com/homepage.html
```

die Anforderung lautet wie folgt:

```
GET /homepage.html HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com
```

Example — CNAME Alias-Methode

Um diese Methode zu verwenden, müssen Sie Ihren DNS Namen als CNAME Alias für konfigurieren `bucket-name.s3.us-east-1.amazonaws.com`. Weitere Informationen finden Sie unter [Amazon S3 URLs mit CNAME Datensätzen anpassen](#).

In diesem Beispiel wird Folgendes verwendet:

- Bucket-Name – `example.com`
- Schlüsselname – `homepage.html`

Das URL ist wie folgt:

```
http://www.example.com/homepage.html
```

Das Beispiel ist wie folgt:

```
GET /homepage.html HTTP/1.1
Host: www.example.com
```

Amazon S3 URLs mit CNAME Datensätzen anpassen

Abhängig von den Anforderungen können Sie die Anzeige von `s3.region-code.amazonaws.com` auf der Website oder im Service gegebenenfalls unterbinden. Wenn Sie beispielsweise Websiteabbilder auf Amazon S3 hosten, bevorzugen Sie möglicherweise `http://images.example.com/` anstelle von `http://images.example.com.s3.us-east-1.amazonaws.com/`. Jeder Bucket mit einem DNS-kompatiblen Namen kann wie folgt referenziert werden: zum `http://BucketName.s3.Region.amazonaws.com/[Filename]`
Beispiel. `http://images.example.com.s3.us-east-1.amazonaws.com/mydog.jpg`

Durch die Verwendung CNAME können Sie einem Amazon S3 S3-Hostnamen zuordnen, sodass der vorherige zu werden URL `http://images.example.com/mydog.jpg` könnte. `images.example.com`

Ihr Bucket-Name muss mit dem CNAME identisch sein. Wenn Sie beispielsweise eine Zuordnung CNAME `images.example.com` zu erstellen `images.example.com.s3.us-east-1.amazonaws.com`, `http://images.example.com.s3.us-east-1.amazonaws.com/filename` werden beide `http://images.example.com/filename` und identisch sein.

Der CNAME DNS Datensatz sollte Ihren Domainnamen mit dem entsprechenden Hostnamen im virtuellen Hosting-Stil verknüpfen. Wenn sich Ihr Bucket-Name und Ihr Domainname beispielsweise in der Region USA Ost (Nord-Virginia) befinden `images.example.com` und Ihr Bucket sich in der Region USA Ost (Nord-Virginia) befindet, sollte der CNAME Datensatz als Alias lauten. `images.example.com.s3.us-east-1.amazonaws.com`

```
images.example.com CNAME    images.example.com.s3.us-east-1.amazonaws.com.
```

Amazon S3 verwendet den Host-Namen, um den Bucket-Namen zu ermitteln. Der Bucket-Name CNAME und der Bucket-Name müssen also identisch sein. Nehmen wir zum Beispiel an, dass Sie `www.example.com` als CNAME für konfiguriert haben `www.example.com.s3.us-east-1.amazonaws.com`. Wenn Sie auf `http://www.example.com` zugreifen, empfängt Amazon S3 eine Anforderung, die in etwa wie folgt aussieht:

Example

```
GET / HTTP/1.1
Host: www.example.com
Date: date
Authorization: signatureValue
```

Amazon S3 sieht nur den ursprünglichen Hostnamen `www.example.com` und weiß nichts von der CNAME Zuordnung, die zur Lösung der Anfrage verwendet wurde.

Sie können jeden Amazon S3 S3-Endpunkt in einem CNAME Alias verwenden. `s3.ap-southeast-1.amazonaws.com` Kann beispielsweise in CNAME Aliassen verwendet werden. Weitere Informationen zu Endpunkten finden Sie unter [Anforderungsendpunkte](#). Informationen zum Erstellen einer statischen Website mit einer benutzerdefinierten Domäne finden Sie unter [Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne](#).

⚠ Important

Wenn Sie custom URLs with verwenden CNAMEs, müssen Sie sicherstellen, dass für jeden CNAME von Ihnen konfigurierten Alias-Datensatz ein passender Bucket vorhanden ist.

Wenn Sie beispielsweise DNS Einträge für Webinhalte erstellen `www.example.com` und diese mit S3 veröffentlichen `login.example.com` möchten, müssen Sie sowohl Buckets `www.example.com` als auch erstellen. `login.example.com`

Wenn ein Aliaseintrag CNAME oder ein Alias konfiguriert ist, der auf einen S3-Endpunkt ohne passenden Bucket verweist, kann jeder AWS Benutzer diesen Bucket erstellen und Inhalte unter dem konfigurierten Alias veröffentlichen, auch wenn die Eigentümerschaft nicht identisch ist.

Aus demselben Grund empfehlen wir, dass Sie beim Löschen eines Buckets den entsprechenden Alias CNAME oder Alias ändern oder entfernen.

Verknüpfen eines Host-Namens mit einem Amazon-S3-Bucket

So verknüpfen Sie mithilfe eines Alias einen Hostnamen mit einem Amazon S3-Bucket CNAME

1. Wählen Sie einen Host-Namen aus, der zu einer von Ihnen kontrollierten Domäne gehört.

Dieses Beispiel verwendet die Unterdomäne `images` der Domäne `example.com`.

2. Erstellen Sie einen Bucket, der dem Host-Namen entspricht.

In diesem Beispiel lauten der Host- und der Bucket-Name `images.example.com`. Der Bucket-Name muss exakt mit dem Host-Namen übereinstimmen.

3. Erstellen Sie einen CNAME DNS Datensatz, der den Hostnamen als Alias für den Amazon S3 S3-Bucket definiert.

Beispielsweise:

```
images.example.com CNAME images.example.com.s3.us-west-2.amazonaws.com
```

⚠ Important

Aus Gründen der Weiterleitung von Anfragen muss der CNAME DNS Datensatz genau so definiert werden, wie im vorherigen Beispiel gezeigt. Andernfalls kann sich trotz richtig erscheinender Funktion unvorhersehbares Verhalten ergeben.

Das Verfahren zur Konfiguration von CNAME DNS Datensätzen hängt von Ihrem DNS Server oder DNS Anbieter ab. Spezifische Informationen finden Sie in Ihrer Serverdokumentation oder erhalten Sie von Ihrem Anbieter.

Einschränkungen

SOAP Support Over HTTP ist veraltet, aber SOAP immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstützt SOAP. Anstatt zu verwenden SOAP, empfehlen wir, entweder den REST API oder den zu verwenden AWS SDKs.

Abwärtskompatibilität

In den folgenden Abschnitten werden verschiedene Aspekte der Amazon S3 S3-Abwärtskompatibilität behandelt, die sich auf Anfragen im Pfadstil und auf virtuell gehostete Anfragen beziehen. URL

Legacy-Endpunkte

Einige Regionen unterstützen Legacy-Endpunkte. Möglicherweise finden Sie diese Endpunkte in Ihren Serverzugriffsprotokollen oder -protokollen. AWS CloudTrail Weitere Informationen finden Sie im folgenden Abschnitt. Eine vollständige Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter [Endpunkte und Kontingente von Amazon S3](#) in der Allgemeine Amazon Web Services-Referenz.

Important

Obwohl Sie möglicherweise Legacy-Endpunkte in Ihren Protokollen sehen, empfehlen wir Ihnen, immer die Standardendpunktsyntax für den Zugriff auf Ihre Buckets zu verwenden. Verwenden Sie im virtuellen Hosted-Stil von Amazon S3 das folgende Format: URLs

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

In Amazon S3 URLs verwendet path-style das folgende Format:

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

S3-Region

Einige ältere Amazon-S3-Regionen unterstützen Endpunkte, die einen Bindestrich (-) zwischen s3 und der Region (z. B. s3-us-west-2) anstelle eines Punkts (z. B. s3.us-west-2) enthalten. Wenn sich Ihr Bucket in einer dieser Regionen befindet, wird in Ihren Serverzugriffsprotokollen oder CloudTrail -protokollen möglicherweise das folgende EndpunktfORMAT angezeigt:

```
https://bucket-name.s3-region-code.amazonaws.com
```

In diesem Beispiel lautet der Bucket-Name amzn-s3-demo-bucket1 und die Region ist US West (Oregon):

```
https://amzn-s3-demo-bucket1.s3-us-west-2.amazonaws.com
```

Globaler Legacy-Endpunkt

Für einige Regionen können Sie den globalen Legacy-Endpunkt verwenden, um Anforderungen zu erstellen, die keinen regionsspezifischen Endpunkt angeben. Der globale Legacy-Endpunkt lautet wie folgt:

```
bucket-name.s3.amazonaws.com
```

In Ihren Serverzugriffsprotokollen oder CloudTrail -protokollen sehen Sie möglicherweise Anfragen, die den alten globalen Endpunkt verwenden. In diesem Beispiel lautet der Bucket-Name amzn-s3-demo-bucket1 und der globale Legacy-Endpunkt ist folgender:

```
https://amzn-s3-demo-bucket1.s3.amazonaws.com
```

Anforderungen im virtuellen Hosting-Format für USA Ost (Nord-Virginia)

Anforderungen, die über den globalen Legacy-Endpunkt gesendet werden, werden standardmäßig an USA Ost (Nord-Virginia) weitergeleitet. Daher wird manchmal der ältere globale Endpunkt anstelle des regionalen Endpunkts für USA Ost (Nord-Virginia) verwendet. Wenn Sie einen Bucket in USA Ost (Nord-Virginia) erstellen und den globalen Endpunkt verwenden, leitet Amazon S3 Ihre Anfrage standardmäßig an diese Region weiter.

Anforderungen im virtuellem Hosting-Format für andere Regionen

Der globale Legacy-Endpunkt wird auch für Anforderungen im virtuellen Hosting-Format in anderen unterstützten Regionen verwendet. Wenn Sie einen Bucket in einer Region erstellen, die vor dem 20.

März 2019 gestartet wurde, und den alten globalen Endpunkt verwenden, aktualisiert Amazon S3 den DNS Datensatz, um die Anfrage an den richtigen Standort umzuleiten, was einige Zeit dauern kann. In der Zwischenzeit wird die Standardregel angewendet und Ihre Anfrage im virtuellen Hosting-Format wird an die Region USA Ost (Nord-Virginia) weitergeleitet. Amazon S3 leitet es dann mit einer temporären HTTP 307-Weiterleitung in die richtige Region weiter.

Bei S3-Buckets in Regionen, die nach dem 20. März 2019 gestartet wurden, leitet der DNS Server Ihre Anfrage nicht direkt an den Ort weiter, an AWS-Region dem sich Ihr Bucket befindet. Stattdessen wird der Fehler HTTP 400 Bad Request zurückgegeben. Weitere Informationen finden Sie unter [Senden von Anforderungen](#).

Anforderungen im Pfadformat

Für die Region USA Ost (Nord-Virginia) können Sie den globalen Legacy-Endpunkt für Anforderungen im Pfadformat verwenden.

Für alle anderen Regionen erfordert die Syntax im Pfadformat, dass beim Zugriff auf einen Bucket der regionsspezifische Endpunkt verwendet werden muss. Wenn Sie versuchen, auf einen Bucket mit dem alten globalen Endpunkt oder einem anderen Endpunkt als dem für die Region zuzugreifen, in der sich der Bucket befindet, erhalten Sie den HTTP Antwortcode 307 Temporärer Umleitungsfehler und eine Meldung, die angibt, dass URI für Ihre Ressource der richtige ist. Wenn Sie beispielsweise `https://s3.amazonaws.com/bucket-name` für einen Bucket verwenden, der in der Region USA West (Oregon) erstellt wurde, erhalten Sie den Fehler HTTP 307 Temporary Redirect.

Anforderungsumleitung und die REST-API

Themen

- [Umleitungen und HTTP-Benutzeragenten](#)
- [Umleitungen und 100 Continue](#)
- [Beispiel für eine Umleitung](#)

In diesem Abschnitt wird beschrieben, wie HTTP-Umleitungen unter Verwendung der Amazon-S3-REST-API verarbeitet werden. Allgemeine Informationen zu Amazon-S3-Weiterleitungen finden Sie unter [Senden von Anforderungen](#) in der Amazon-Simple-Storage-Service-API-Referenz.

Umleitungen und HTTP-Benutzeragenten

Programme, die die Amazon S3 REST-API verwenden, sollten Umleitungen auf der Anwendungsschicht oder auf der HTTP-Ebene verarbeiten. Es können viele HTTP-Client-

Bibliotheken und Benutzeragenten konfiguriert werden, um Umleitungen automatisch korrekt zu verarbeiten. Viele andere beinhalten jedoch fehlerhafte oder unvollständige Implementierungen für Umleitungen.

Bevor Sie sich darauf verlassen, dass eine Bibliothek die Umleitungsanfrage erfüllt, testen Sie die folgenden Fälle:

- Überprüfen Sie, ob alle HTTP-Anfrageheader korrekt in die Umleitungsanfrage aufgenommen wurden (die zweite Anfrage nach Empfang einer Umleitung), einschließlich der HTTP-Standards, wie beispielsweise Autorisierung und Datum.
- Überprüfen Sie, ob auch andere als GET-Umleitungen ordnungsgemäß funktionieren, wie beispielsweise PUT und DELETE.
- Überprüfen Sie, ob PUT-Anfragen den Umleitungen ordnungsgemäß folgen.
- Überprüfen Sie, ob PUT-anfragenden Umleitungen ordnungsgemäß folgen, wenn die Antwort auf 100 continue lang dauert.

HTTP-Benutzeragenten, die streng konform zu RFC 2616 arbeiten, fordern möglicherweise eine explizite Bestätigung, bevor sie einer Umleitung folgen, wenn die HTTP-Anfragemethode nicht GET oder HEAD ist. Im Allgemeinen ist es sicher, von Amazon S3 erstellten Umleitungen automatisch zu folgen, weil das System nur Umleitungen auf Hosts innerhalb der Domäne amazonaws.com ausgibt, und die Wirkung der umgeleiteten Anfrage dieselbe ist wie die der ursprünglichen Anforderung.

Umleitungen und 100 Continue

Um die Verarbeitung von Umleitungen zu vereinfachen, die Effizienzen zu verbessern und die Kosten zu vermeiden, die beim doppelten Versenden eines umgeleiteten Anfragerumpfs entstehen, konfigurieren Sie Ihre Anwendung so, dass sie 100 continues für PUT-Operationen verwendet. Wenn Ihre Anwendung 100 continue verwendet, sendet sie den Anfragerumpf erst dann, wenn sie eine Bestätigung erhält. Wird die Nachricht basierend auf den Header abgewiesen, wird der Rumpf der Meldung nicht gesendet. Weitere Informationen zu 100-continue finden Sie unter [RFC 2616 Abschnitt 8.2.3](#).

Note

Laut RFC 2616 sollten Sie nicht unendlich lange warten, bevor Sie den Anfragerumpf senden, wenn Sie `Expect: Continue` für einen unbekanntes HTTP-Server verwenden. Der Grund dafür ist, dass einige HTTP-Server 100 continue nicht erkennen. Amazon S3 erkennt jedoch nicht, ob Ihre Anforderung ein `Expect: Continue` enthält, und reagiert mit einem

vorläufigen 100-continue-Status oder einem endgültigen Statuscode. Darüber hinaus tritt kein Umleitungscode auf, nachdem die provisorische 100-continue-Genehmigung empfangen wurde. Dies hilft Ihnen, den Empfang einer Umleitungsantwort zu vermeiden, wenn Sie noch den Anfragerumpf schreiben.

Beispiel für eine Umleitung

Dieser Abschnitt bietet ein Beispiel für eine Client/Server-Interaktion mit HTTP-Umleitungen und 100 continue.

Nachfolgend finden Sie ein Beispiel für ein PUT in den quotes.s3.amazonaws.com-Bucket.

```
PUT /nelson.txt HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Mon, 15 Oct 2007 22:18:46 +0000

Content-Length: 6
Expect: 100-continue
```

Amazon S3 gibt Folgendes zurück:

```
HTTP/1.1 307 Temporary Redirect
Location: http://quotes.s3-4c25d83b.amazonaws.com/nelson.txt?rk=8d47490b
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Mon, 15 Oct 2007 22:18:46 GMT

Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>TemporaryRedirect</Code>
  <Message>Please re-send this request to the
  specified temporary endpoint. Continue to use the
  original request endpoint for future requests.
</Message>
  <Endpoint>quotes.s3-4c25d83b.amazonaws.com</Endpoint>
  <Bucket>quotes</Bucket>
</Error>
```


Der Client folgt der Umleitungsantwort und gibt eine neue Anfrage an den temporären Endpunkt `quotes.s3-4c25d83b.amazonaws.com` aus.

```
PUT /nelson.txt?rk=8d47490b HTTP/1.1
Host: quotes.s3-4c25d83b.amazonaws.com
Date: Mon, 15 Oct 2007 22:18:46 +0000

Content-Length: 6
Expect: 100-continue
```

Amazon S3 gibt ein 100 continue zurück, das darauf hinweist, dass der Client damit fortfahren soll, den Anforderungstext zu senden.

```
HTTP/1.1 100 Continue
```

Der Client sendet den Anfragerumpf.

```
ha ha\n
```

Amazon S3 gibt die endgültige Antwort zurück:

```
HTTP/1.1 200 OK
Date: Mon, 15 Oct 2007 22:18:48 GMT

ETag: "a2c8d6b872054293afd41061e93bc289"
Content-Length: 0
Server: AmazonS3
```

Entwickeln mit Amazon S3 über die AWS CLI

Befolgen Sie diese Schritte zum Herunterladen und Konfigurieren der AWS Command Line Interface (AWS CLI).

Eine Liste der Amazon-S3-AWS CLI-Befehle finden Sie auf den folgenden Seiten der AWS CLI-Befehlsreferenz:

- [S3](#)
- [s3api](#)

- [s3control](#)

Note

Services in AWS, wie z. B. Amazon S3, erfordern beim Zugriff die Eingabe von Anmeldeinformationen. Der Service kann dann feststellen, ob Sie über die Berechtigung für den Zugriff auf seine Ressourcen verfügen. Für die Konsole müssen Sie Ihr Passwort eingeben. Sie können Zugriffsschlüssel für Ihr AWS-Konto erstellen, um auf die AWS CLI oder die API zuzugreifen. Wir raten Ihnen jedoch davon ab, mittels der Anmeldeinformationen für Ihr AWS-Konto auf AWS zuzugreifen. Stattdessen empfehlen wir, AWS Identity and Access Management (IAM) zu verwenden. Erstellen Sie einen IAM-Benutzer und fügen Sie den Benutzer zu einer IAM-Gruppe mit Administrator-Berechtigungen hinzu. Anschließend gewähren Sie dem von Ihnen erstellten IAM-Benutzer administrative Berechtigungen. Sie können dann mithilfe einer speziellen URL und mit den Anmeldeinformationen für den IAM-Benutzer auf AWS zugreifen. Weitere Anweisungen finden Sie unter [Erstellen Ihrer ersten IAM-Benutzer- und Administratorengruppe](#) im IAM-Benutzerhandbuch.

Einrichten von AWS CLI

1. Herunterladen und Konfigurieren von AWS CLI. Eine Anleitung finden Sie unter den folgenden Themen im AWS Command Line Interface-Benutzerhandbuch:
 - [Einrichtung der AWS Command Line Interface](#)
 - [Konfigurieren von AWS Command Line Interface](#)
2. Fügen Sie ein benanntes Profil für den Administratorbenutzer in der AWS CLI-Konfigurationsdatei hinzu. Verwenden Sie dieses Profil beim Ausführen von AWS CLI-Befehlen. Weitere Informationen finden Sie unter [Benannte Profile für die AWS CLI](#) im AWS Command Line Interface-Benutzerhandbuch.

```
[adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```

Eine Liste der verfügbaren AWS-Regionen finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

- Überprüfen Sie die Einrichtung, indem Sie die folgenden Befehle in die Befehlszeile eingeben.
 - Führen Sie den `help`-Befehl aus, um zu überprüfen, ob die AWS CLI auf Ihrem Computer installiert ist:

```
aws help
```

- Führen Sie einen Befehl `s3` mit den Anmeldeinformationen `adminuser` aus, die Sie soeben erstellt haben. Fügen Sie hierzu Ihrem Befehl den Parameter `--profile` hinzu, um den Profilenames anzugeben. In diesem Beispiel listet der Befehl `ls` Buckets in Ihrem Konto auf. Die AWS CLI verwendet `adminuser`-Anmeldeinformationen, um die Anforderung zu authentifizieren.

```
aws s3 ls --profile adminuser
```

Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

Note


Sie können es AWS Amplify für die end-to-end Full-Stack-Entwicklung von Web- und mobilen Apps verwenden. Amplify Storage integriert Dateispeicher- und Verwaltungsfunktionen nahtlos in Frontend-Web- und Mobilanwendungen, die auf Amazon S3 aufbauen. Weitere Informationen finden Sie unter [Speicher](#) im Amplify-Benutzerhandbuch.

Verwenden Sie diesen Dienst mit einem SDK AWS

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK for C++	AWS SDK for C++ Code-Beispiele
AWS CLI	AWS CLI Codebeispiele
AWS SDK for Go	AWS SDK for Go Codebeispiele
AWS SDK for Java	AWS SDK for Java Codebeispiele
AWS SDK for JavaScript	AWS SDK for JavaScript Codebeispiele
AWS SDK for Kotlin	AWS SDK for Kotlin Codebeispiele
AWS SDK for .NET	AWS SDK for .NET Codebeispiele
AWS SDK for PHP	AWS SDK for PHP Codebeispiele
AWS Tools for PowerShell	Tools für PowerShell Codebeispiele
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) Codebeispiele
AWS SDK for Ruby	AWS SDK for Ruby Codebeispiele
AWS SDK for Rust	AWS SDK for Rust Codebeispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Codebeispiele
AWS SDK for Swift	AWS SDK for Swift Codebeispiele

Weitere Beispiele speziell für diesen Service finden Sie unter [Codebeispiele für Amazon S3 mit AWS SDKs](#).

 Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Feedback geben auswählen.

SDK-Programmierschnittstellen

Jedes AWS SDK bietet eine oder mehrere programmatische Schnittstellen für die Arbeit mit Amazon S3. Jedes SDK bietet eine Low-Level-Schnittstelle für Amazon S3 mit Methoden, die API-Vorgängen sehr ähnlich sind. Einige SDKs bieten High-Level-Schnittstellen für Amazon S3, bei denen es sich um Abstraktionen handelt, die allgemeine Anwendungsfälle vereinfachen sollen.

Wenn Sie beispielsweise einen mehrteiligen Upload mithilfe der API-Operationen auf niedriger Ebene durchführen, müssen Sie einen Vorgang verwenden, um den Upload zu starten, einen weiteren Vorgang, um Teile hochzuladen, und einen letzten Vorgang, um den Upload abzuschließen. Ein mehrteiliger Upload-API-Vorgang auf hoher Ebene ermöglicht es Ihnen, alle für den Upload erforderlichen Vorgänge in einem einzigen API-Aufruf durchzuführen. Beispiele finden Sie unter [Hochladen eines Objekts mit Multipart-Upload](#).

API-Operationen auf niedriger Ebene ermöglichen eine bessere Kontrolle über den Upload. Wir empfehlen Ihnen, API-Operationen auf niedriger Ebene zu verwenden, wenn Sie Uploads pausieren und fortsetzen, die Teilgrößen während des Uploads variieren oder Uploads starten müssen, wenn Sie die Größe der Daten nicht im Voraus kennen.

Angabe der Signature-Version in der Anforderungs-Authentifizierung

Amazon S3 unterstützt in den meisten AWS-Regionen Fällen nur AWS Signature Version 4. In einigen älteren Versionen AWS-Regionen unterstützt Amazon S3 sowohl Signature Version 4 als auch Signature Version 2. Signature Version 2 wird jedoch deaktiviert (veraltet). Weitere Informationen zum Ende der Unterstützung für Signature Version 2 finden Sie unter [AWS Signature Version 2 wurde für Amazon S3 deaktiviert \(veraltet\)](#).

Eine Liste aller Amazon-S3-Regionen und der von ihnen unterstützten Signature-Versionen finden Sie unter [Regions and Endpoints \(Regionen und Endpunkte\)](#) in der Allgemeinen AWS -Referenz.

Für alle AWS-Regionen verwenden AWS SDKs standardmäßig Signature Version 4 zur Authentifizierung von Anfragen. Wenn Sie AWS SDKs verwenden, die vor Mai 2016 veröffentlicht wurden, müssen Sie möglicherweise Signature Version 4 anfordern, wie in der folgenden Tabelle dargestellt.

SDK	Anforderung von Signature Version 4 für die Anforderungs-Authentifizierung
AWS CLI	Für das Standardprofil führen Sie den folgenden Befehl aus:

SDK	Anforderung von Signature Version 4 für die Anforderungs-Authentifizierung
	<pre>\$ aws configure set default.s3.signature_version s3v4</pre> <p>Für ein benutzerdefiniertes Profil führen Sie den folgenden Befehl aus:</p> <pre>\$ aws configure set profile.your_profile_name.s3.signature_version s3v4</pre>
Java-SDK	Fügen Sie Ihrem Code Folgendes hinzu: <pre>System.setProperty(SDKGlobalConfiguration.ENABLE_S3_SIGV4_SYSTEM_PROPERTY, "true");</pre> Oder geben Sie in der Befehlszeile Folgendes an: <pre>-Dcom.amazonaws.services.s3.enableV4</pre>
JavaScript SDK	Setzen Sie den <code>signatureVersion</code> -Parameter auf <code>v4</code> , wenn Sie den Client konstruieren: <pre>var s3 = new AWS.S3({signatureVersion: 'v4'});</pre>

SDK	Anforderung von Signature Version 4 für die Anforderungs-Authentifizierung
PHP SDK	<p>Setzen Sie den Parameter <code>signature</code> auf <code>v4</code>, wenn Sie den Amazon S3 Service-Client für PHP SDK v2 erstellen:</p> <pre data-bbox="597 394 1507 674"><?php \$client = S3Client::factory(['region' => 'YOUR-REGION', 'version' => 'latest', 'signature' => 'v4']);</pre> <p>Setzen Sie den Parameter <code>signature_version</code> bei Verwendung des PHP SDK v3 im Rahmen der Erstellung des Amazon S3 Service-Clients auf <code>v4</code>:</p> <pre data-bbox="597 877 1507 1157"><?php \$s3 = new Aws\S3\S3Client(['version' => '2006-03-01', 'region' => 'YOUR-REGION', 'signature_version' => 'v4']);</pre>
Python-Boto SDK	<p>Geben Sie Folgendes in der boto-Standardkonfigurationsdatei an:</p> <pre data-bbox="597 1318 1507 1388">[s3] use-sigv4 = True</pre>

SDK	Anforderung von Signature Version 4 für die Anforderungs-Authentifizierung
Ruby SDK	<p>Ruby SDK – Version 1: Setzen Sie den <code>:s3_signature_version</code> -Parameter auf <code>:v4</code>, wenn Sie den Client konstruieren:</p> <pre>s3 = AWS::S3::Client.new(:s3_signature_version => :v4)</pre> <p>Ruby SDK – Version 3: Setzen Sie den <code>signature_version</code> -Parameter auf <code>v4</code>, wenn Sie den Client konstruieren:</p> <pre>s3 = Aws::S3::Client.new(signature_version: 'v4')</pre>
.NET SDK	<p>Fügen Sie den folgenden Code hinzu, bevor Sie den Amazon-S3-Client erstellen:</p> <pre>AWSConfigsS3.UseSignatureVersion4 = true;</pre> <p>Oder fügen Sie der Konfigurationsdatei Folgendes hinzu:</p> <pre><appSettings> <add key="AWS.S3.UseSignatureVersion4" value="true" /> </appSettings></pre>

AWS Signature Version 2 wurde für Amazon S3 deaktiviert (veraltet)

Signature Version 2 wird unter Amazon S3 deaktiviert (veraltet). Danach akzeptiert Amazon S3 nur noch API-Anforderungen, die mithilfe der Signature Version 4 signiert wurden.

Dieser Abschnitt umfasst Antworten auf häufig gestellte Fragen im Bezug auf das Supportende für Signature Version 2.

Um was handelt es sich bei der Signature Version 2/4 und was versteht man unter dem Signieren einer Anforderung?

Der Signiervorgang mittels Signature Version 2 oder Signature Version 4 authentifiziert Ihre API-Anforderungen in Amazon S3. Durch das Signieren von Anforderungen ist Amazon S3 in der Lage, den Absender der Anforderung zu identifizieren und Ihre Anforderungen vor Angreifern zu schützen.

Weitere Informationen zum Signieren von AWS Anfragen finden Sie unter [Signieren von AWS API-Anfragen](#) in der Allgemeine AWS-Referenz

Welche Aktualisierung nehmen Sie vor?

Derzeit unterstützen wir Amazon S3 API-Anforderungen, die mithilfe von Signature Version 2 und Signature Version 4 signiert wurden. Später akzeptiert Amazon S3 nur noch Anforderungen, die mit Signature Version 4 signiert wurden.

Weitere Informationen zum Signieren von AWS Anfragen finden Sie unter [Änderungen in Signature Version 4](#) in der Allgemeine AWS-Referenz.

Warum nehmen Sie die Aktualisierung vor?

Signature Version 4 bietet mehr Sicherheit, da hierbei im Gegensatz zum geheimen Zugriffsschlüssel ein Signaturschlüssel verwendet wird. Signature Version 4 wird derzeit in allen Versionen unterstützt AWS-Regionen, während Signature Version 2 nur in Regionen unterstützt wird, die vor Januar 2014 eingeführt wurden. Dank dieser Aktualisierung sind wir in der Lage, in allen Regionen eine einheitlichere Erfahrung bereitzustellen.

Wie kann ich sicherstellen, dass ich Signature Version 4 verwende und welche Aktualisierungen benötige ich?

Normalerweise entscheidet das Tool oder das SDK auf Clientseite, welche Signature Version zum Signieren der Anforderungen verwendet wird. Standardmäßig verwenden die neuesten Versionen unserer AWS SDKs Signature Version 4. Setzen Sie sich mit dem entsprechenden Supportteam für Ihre Drittanbietersoftware in Verbindung, um zu erfahren, welche Version Sie benötigen. Um direkte REST-Aufrufe an Amazon S3 zu senden, müssen Sie Ihre Anwendung so anpassen, dass der Signiervorgang mithilfe von Signature Version 4 ausgeführt wird.

Informationen darüber, welche Version der AWS SDKs bei der Umstellung auf Signature Version 4 verwendet werden sollte, finden Sie unter [Von Signature Version# 2 zur Signature Version 4 wechseln](#).

Weitere Informationen zur Verwendung von Signature Version 4 mit der Amazon S3 REST-API finden Sie unter [Authenticating Requests \(Authentifizieren von Anforderungen\) \(AWS Signature Version 4\)](#) in der API-Referenz zu Amazon Simple Storage Service.

Was passiert, wenn ich keine Aktualisierung durchführe?

Anforderungen, mit Signature Version 2 signiert wurden und zu einem späteren Zeitpunkt gesendet werden, können dann nicht mehr mit Amazon S3 authentifiziert werden. Anforderer sehen eine Fehlermeldung, dass die Anforderung mit Signature Version 4 signiert sein muss.

Sollte ich Änderungen vornehmen, selbst wenn ich eine vorsignierte URL verwende, die es erforderlich macht, dass ich für mehr als sieben Tage signiere?

Wenn sie eine vorsignierte URL verwenden, für die Sie für mehr als sieben Tage signieren müssen, ist keine Maßnahme erforderlich. Sie können AWS Signature Version 2 weiterhin verwenden, um die vorsignierte URL zu signieren und zu authentifizieren. Wir werden uns darum kümmern und zusätzliche Einzelheiten zur Migration zu Signature Version 4 für ein vorab signiertes URL-Szenario bereitstellen.

Weitere Infos

- Weitere Informationen zur Verwendung von Signature Version 4 finden Sie unter [Signieren von AWS API-Anfragen](#).
- Sehen Sie sich die Änderungen zwischen Signature Version 2 und Signature Version 4 unter [Änderungen in Signature Version 4](#) an.
- Sehen Sie sich den Beitrag [AWS Signature Version 4 als Ersatz für AWS Signature Version 2 zum Signieren von Amazon S3 S3-API-Anfragen](#) in den AWS Foren an.
- Wenden Sie sich bei weiteren Fragen oder Bedenken unter [AWS Support](#) an uns.

Von Signature Version# 2 zur Signature Version 4 wechseln.

Wenn Sie aktuell Signature Version 2 für die Authentifizierung der API-Anforderungen in Amazon S3 verwenden, sollten Sie möglichst bald auf Signature Version 4 umwechseln. Das Ende der Unterstützung für Signature Version 2 wird unter beschriebene [AWS Signature Version 2 wurde für Amazon S3 deaktiviert \(veraltet\)](#).

Weitere Informationen zur Verwendung von Signature Version 4 mit der Amazon S3 REST-API finden Sie unter [Authenticating Requests \(Authentifizieren von Anforderungen\) \(AWS Signature Version 4\)](#) in der API-Referenz zu Amazon Simple Storage Service.

Die folgende Liste stellt die SDKs mit der erforderlichen Mindestversion zur Verwendung von Signature Version 4 (SigV4) dar. Wenn Sie vorsignierte URLs mit den AWS Java-, JavaScript (Node.js) oder Python-SDKs (Boto/CLI) verwenden, müssen Sie in der Client-Konfiguration die richtige AWS-Region und Signature Version 4 festlegen. Weitere Informationen zum Einrichten von SigV4 in der Client-Konfiguration finden Sie unter [Angabe der Signature-Version in der Anforderungs-Authentifizierung](#).

Wenn Sie diese SDK/ dieses Produkt verwenden	Aktualisieren Sie auf diese SDK-Version	Bei Verwendung von Sigv4 Codeänderung am Client erforderlich?	Link zur SDK-Dokumentation
AWS SDK for Java v1	Upgrade auf Java 1.11.201+ oder v2.	Ja	Angabe der Signature-Version in der Anforderungs-Authentifizierung
AWS SDK for Java v2	Keine SDK-Aktualisierung erforderlich.	Nein	AWS SDK for Java
AWS SDK for .NET v1	Auf 3.1.10 oder höher aktualisieren.	Ja	AWS SDK for .NET
AWS SDK for .NET v2	Auf 3.1.10 oder höher aktualisieren.	Nein	AWS SDK for .NET v2

Wenn Sie diese SDK/ dieses Produkt verwenden	Aktualisieren Sie auf diese SDK-Version	Bei Verwendung von Sigv4 Codeänderung am Client erforderlich?	Link zur SDK-Dokumentation
AWS SDK for .NET v3	Auf 3.3.0.0 oder höher aktualisieren.	Ja	AWS SDK for .NET v3
AWS SDK for JavaScript v1	Auf 2.68.0 oder höher aktualisieren.	Ja	AWS SDK for JavaScript
AWS SDK for JavaScript v2	Auf 2.68.0 oder höher aktualisieren.	Ja	AWS SDK for JavaScript
AWS SDK for JavaScript v3	Derzeit ist keine weitere Maßnahme erforderlich. Im Q3 2019 auf Hauptversion V3 aktualisieren.	Nein	AWS SDK for JavaScript

Wenn Sie diese SDK/ dieses Produkt verwenden	Aktualisieren Sie auf diese SDK-Version	Bei Verwendung von Sigv4 Codeänderung am Client erforderlich?	Link zur SDK-Dokumentation
AWS SDK for PHP v1	Empfohlen wird ein Upgrade auf die aktuelle Version von PHP oder mindestens v2.7.4 und der Wert v4 für den Signature-Parameter in der Konfiguration des S3-Clients.	Ja	AWS SDK for PHP

Wenn Sie diese SDK/ dieses Produkt verwenden	Aktualisieren Sie auf diese SDK-Version	Bei Verwendung von Sigv4 Codeänderung am Client erforderlich?	Link zur SDK-Dokumentation
AWS SDK for PHP v2	Empfohlen wird ein Upgrade auf die aktuelle Version von PHP oder mindestens v2.7.4 und der Wert v4 für den Signature-Parameter in der Konfiguration des S3-Clients.	Nein	AWS SDK for PHP
AWS SDK for PHP v3	Keine SDK-Aktualisierung erforderlich.	Nein	AWS SDK for PHP
Boto2	Auf Boto2 v2.49.0 aktualisieren.	Ja	Boto 2-Aktualisierung
Boto3	Auf 1.5.71 (Botocore), 1.4.6 (Boto3) aktualisieren.	Ja	Boto 3 - AWS SDK für Python

Wenn Sie diese SDK/ dieses Produkt verwenden	Aktualisieren Sie auf diese SDK-Version	Bei Verwendung von Sigv4 Codeänderung am Client erforderlich?	Link zur SDK-Dokumentation
AWS CLI	Auf 1.11.108 aktualisieren.	Ja	AWS Command Line Interface
AWS CLI v2 (Vorschau)	Keine SDK-Aktualisierung erforderlich.	Nein	AWS Command Line Interface Version 2
AWS SDK for Ruby v1	Auf Ruby V3 aktualisieren.	Ja	Ruby V3 für AWS
AWS SDK for Ruby v2	Auf Ruby V3 aktualisieren.	Ja	Ruby V3 für AWS
AWS SDK for Ruby v3	Keine SDK-Aktualisierung erforderlich.	Nein	Ruby V3 für AWS
Go	Keine SDK-Aktualisierung erforderlich.	Nein	AWS SDK for Go
C++	Keine SDK-Aktualisierung erforderlich.	Nein	AWS SDK for C++

AWS Tools for Windows PowerShell oder AWS Tools for PowerShell Core

Wenn Sie Modulversionen verwenden, die älter als 3.3.0.0 sind, müssen Sie auf 3.3.0.0 aktualisieren.

Verwenden Sie `Get-Module` cmdlet, um die Versionsinformationen zu erhalten:

```
Get-Module -Name AWSPowershell  
Get-Module -Name AWSPowershell.NetCore
```

Um auf die Version 3.3.0.0 zu aktualisieren, verwenden Sie das cmdlet `Update-Module`:

```
Update-Module -Name AWSPowershell  
Update-Module -Name AWSPowershell.NetCore
```

Für Signature Version 2-Datenverkehr können Sie vorab signierte URLs verwenden, die länger als sieben Tage gültig sind.

Entwickeln mit Amazon S3 unter Verwendung der REST-API

Die Amazon-S3-Architektur ist so ausgelegt, dass sie unabhängig von Programmiersprachen ist und unsere unterstützten Schnittstellen verwendet, um Objekte zu speichern und abzurufen.

Amazon S3 bietet derzeit eine REST-Schnittstelle. Mit REST werden Metadaten in HTTP-Headern zurückgegeben. Wir unterstützen nur HTTP-Anfragen von bis zu 4 KB (ohne den Rumpf), deshalb ist die Menge der von Ihnen bereitgestellten Metadaten begrenzt. Die REST-API ist eine HTTP-Schnittstelle zu Amazon S3. Mit REST verwenden Sie HTTP-Standardanfragen, um Buckets und Objekte zu erstellen, laden oder löschen.

Sie können einen beliebigen Toolkit einsetzen, der HTTP unterstützt, um die REST-API verwenden zu können. Sie können sogar einen Browser verwenden, um Objekte zu laden, wenn diese anonym lesbar sind.

Die REST-API verwendet die HTTP-Standardheader und -Statuscodes, sodass sich Standard-Browser und -Toolkits wie erwartet verhalten. In einigen Bereichen haben wir HTTP um zusätzliche Funktionen erweitert (wir haben beispielsweise Header hinzugefügt, um die Zugriffskontrolle zu unterstützen). In diesen Fällen haben wir alles dafür getan, die neue Funktion so hinzuzufügen, dass sie der standardmäßigen Nutzung von HTTP entsprechen.

Weitere Informationen zum Senden von Anforderungen mit der REST-API finden Sie unter [Anfragen stellen mit dem REST API](#). In den folgenden Themen finden Sie einige Überlegungen, die Sie bei der Verwendung der REST API beachten sollten.

Weitere Informationen zur REST-API von Amazon S3 finden Sie in der [Amazon Simple Storage Service API-Referenz](#).

Themen

- [Weiterleitung von Anforderungen](#)

Weiterleitung von Anforderungen

Programme, die Anfragen für Buckets stellen, die mit der [CreateBucket](#)-API und einer [CreateBucketConfiguration](#) erstellt wurden, müssen Umleitungen unterstützen. Darüber hinaus entstehen möglicherweise für einige Clients Probleme, die DNS TTLs nicht unterstützen.

Dieser Abschnitt beschreibt Weiterleitungs- und DNS-Aspekte, die Sie beim Entwurf Ihres Service oder Ihrer Anwendung für Amazon S3 berücksichtigen sollten.

Anforderungsumleitung und die REST-API

Amazon S3 verwendet das DNS (Domain Name System), um Anfragen an Einrichtungen weiterzuleiten, die sie verarbeiten können. Dieses System arbeitet effektiv, aber es können temporäre Routing-Fehler auftreten. Trifft eine Anfrage am falschen Amazon-S3-Standort ein, reagiert Amazon S3 mit einer temporären Umleitung, die den Auftraggeber anweist, die Anfrage an einen neuen Endpunkt zu senden. Ist eine Anfrage fehlerhaft aufgebaut, verwendet Amazon S3 permanente Umleitungen, um dem Auftraggeber mitzuteilen, wie die Anfrage korrekt ausgeführt wird.

Important

Um diese Funktion nutzen zu können, benötigen Sie eine Anwendung, die Amazon-S3-Umleitungsantworten verarbeiten kann. Die einzige Ausnahme bilden Anwendungen, die ausschließlich mit Buckets arbeiten, die ohne erstellt wurde `<CreateBucketConfiguration>`. Weitere Informationen über Standorteinschränkungen finden Sie unter [Zugreifen auf einen Amazon-S3-Bucket und Auflisten des Buckets](#). Für alle Regionen, die nach dem 20. März 2019 gestartet wurden, gibt Amazon S3 den Fehler „HTTP 400 Bad Request“ zurück, wenn eine Anforderung an einer falschen Amazon-S3-Position empfangen wird.

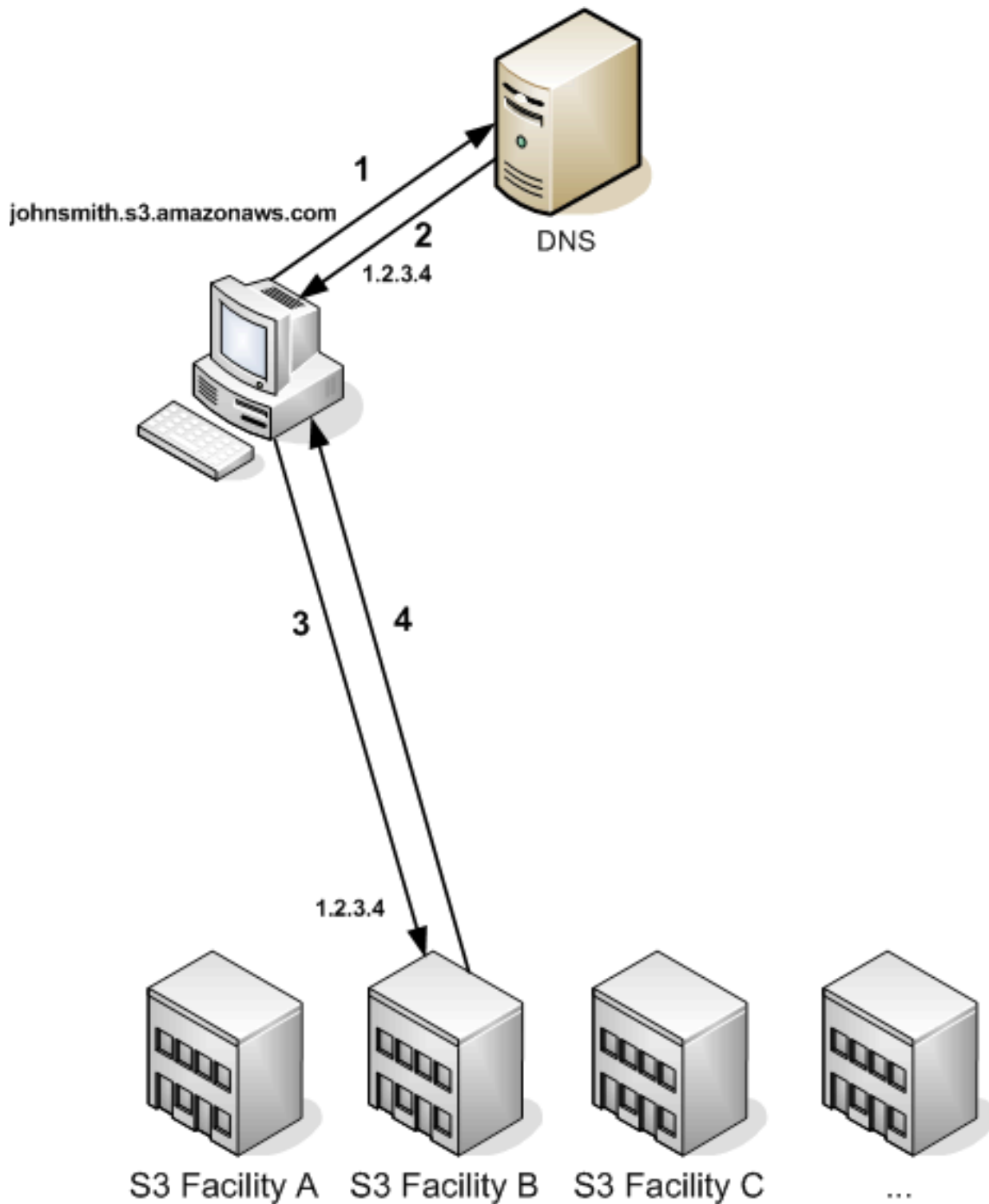
Weitere Informationen zum Aktivieren oder Deaktivieren einer AWS-Region finden Sie unter [AWS-Regionen und -Endpunkte](#) in der Allgemeine AWS-Referenz.

Themen

- [DNS-Weiterleitung](#)
- [Temporäre Anforderungsumleitung](#)
- [Permanente Anforderungsumleitung](#)
- [Beispiele für Anforderungsumleitung](#)

DNS-Weiterleitung

Die DNS-Weiterleitung leitet Anfragen an geeignete Amazon-S3-Einrichtungen weiter. Die folgende Abbildung und Vorgehensweise zeigen ein Beispiel für DNS-Routing.



DNS-Routing-Anforderungsschritte

1. Der Client stellt eine DNS-Anfrage, um ein Objekt in Amazon S3 speichern zu lassen.

2. Der Client erhält eine oder mehrere IP-Adressen für Einrichtungen, die die Anfrage verarbeiten können. In diesem Beispiel ist die IP-Adresse für die Einrichtung B.
3. Der Client stellt eine Anforderung an Amazon-S3-Einrichtung B.
4. Einrichtung B gibt eine Kopie des Objekts an den Kunden zurück.

Temporäre Anforderungsumleitung

Eine temporäre Umleitung ist eine Art Fehlermeldung, die dem Auftraggeber signalisiert, dass er die Anfrage an einen anderen Endpunkt senden sollte. Aufgrund der verteilten Natur von Amazon S3 können Anfragen temporär in die falsche Einrichtung weitergeleitet werden. Dies tritt am wahrscheinlichsten auf, unmittelbar nachdem Buckets erstellt oder gelöscht wurden.

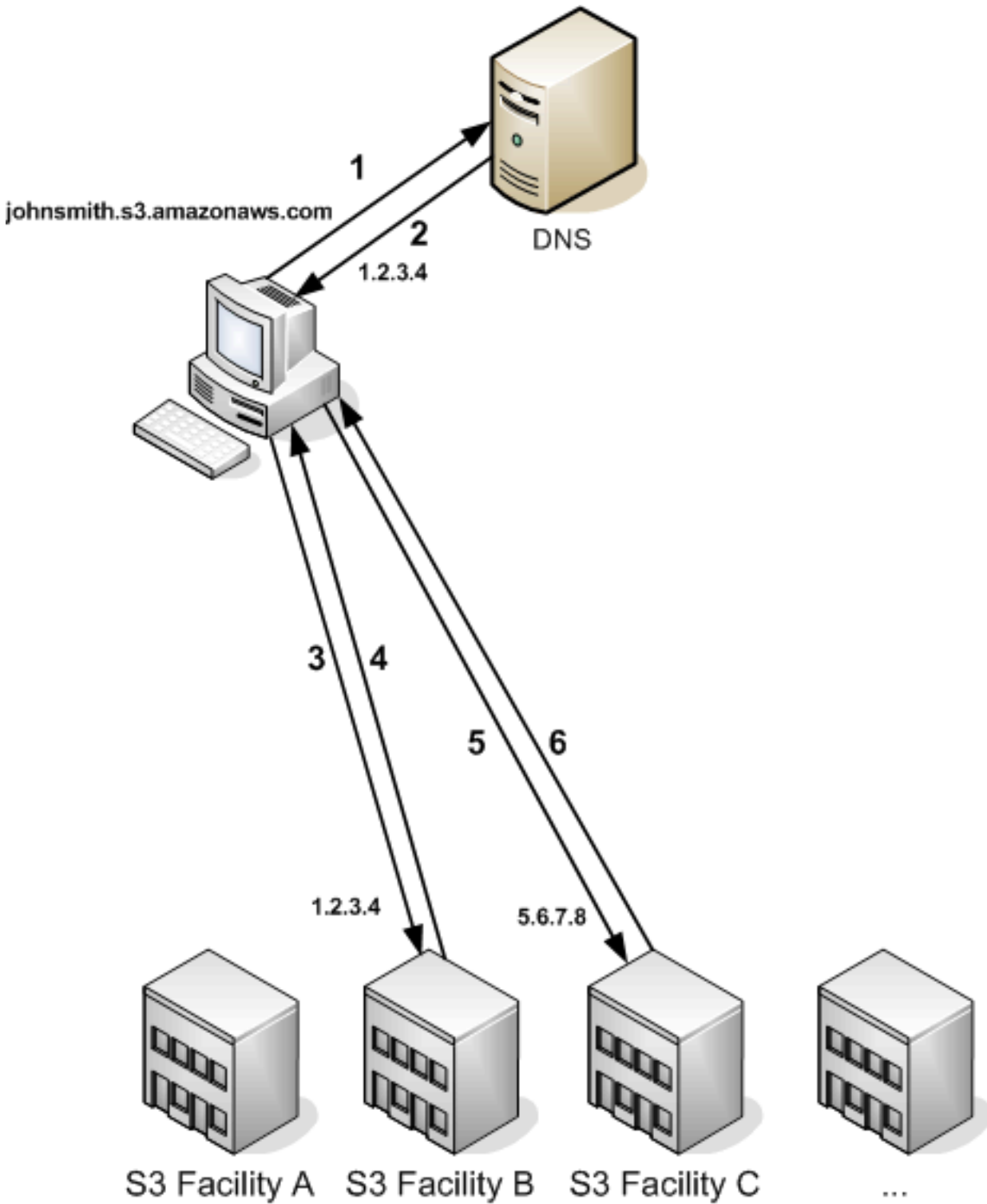
Wenn Sie beispielsweise einen neuen Bucket erstellen und sofort eine Anfrage für den Bucket machen, werden Sie möglicherweise temporär umgeleitet, abhängig von der Standorteinschränkung des Buckets. Wenn Sie den Bucket in der AWS-Region USA Ost (Nord-Virginia) erstellt haben, wird die Umleitung nicht angezeigt, da dies auch der Standard-Amazon-S3-Endpunkt ist.

Wenn der Bucket jedoch in einer anderen Region erstellt wird, gehen alle Anfragen für den Bucket an den Standard-Endpunkt, während der DNS-Eintrag des Buckets propagiert wird. Der Standard-Endpunkt leitet die Anfrage mit einer HTTP 302-Antwort an den richtigen Endpunkt weiter. Temporäre Umleitungen enthalten eine URI zur richtigen Einrichtung, die Sie verwenden können, um die Anfrage sofort erneut zu senden.

Important

Verwenden Sie keinen Endpunkt wieder, der aus einer vorherigen Umleitungsantwort stammt. Das scheint zu funktionieren (manchmal sogar sehr lange), kann aber unvorhersehbare Ergebnisse erzeugen und schlägt irgendwann ohne weitere Mitteilung fehl.

Die folgende Abbildung und Vorgehensweise zeigen ein Beispiel für eine temporäre Umleitung.



Temporäre Umleitungsschritte für Anfragen

1. Der Client stellt eine DNS-Anfrage, um ein Objekt in Amazon S3 speichern zu lassen.

2. Der Client erhält eine oder mehrere IP-Adressen für Einrichtungen, die die Anfrage verarbeiten können.
3. Der Client stellt eine Anforderung an Amazon-S3-Einrichtung B.
4. Einrichtung B gibt eine Umleitung zurück, die angibt, dass das Objekt an Standort C zur Verfügung steht.
5. Der Client sendet Anfrage noch einmal an Einrichtung C.
6. Einrichtung C gibt eine Kopie des Objekts zurück.

Permanente Anforderungsumleitung

Eine permanente Anfrageumleitung zeigt an, dass Ihre Anfrage eine fehlerhafte Adresse für eine Ressource angegeben hat. Permanente Umleitungen treten beispielsweise auf, wenn Sie eine Anfrage mit Angabe eines Pfads für den Zugriff auf einen Bucket verwenden, der mit erstellt wurde `<CreateBucketConfiguration>`. Weitere Informationen finden Sie unter [Zugreifen auf einen Amazon-S3-Bucket und Auflisten des Buckets](#).

Um diese Fehler bei der Entwicklung zu erkennen, enthält diese Art Umleitung keinen HTTP Location-Header, der Ihnen ermöglicht, die Anfrage automatisch an den richtigen Standort zu verfolgen. Weitere Informationen über die Verwendung des richtigen Amazon-S3-Endpunkts finden Sie im resultierenden XML-Fehlerdokument.

Beispiele für Anforderungsumleitung

Im Folgenden finden Sie Beispiele für temporäre Umleitungsantworten.

Temporäre Umleitungsantwort der REST-API

```
HTTP/1.1 307 Temporary Redirect
Location: http://awsexamplebucket1.s3-gztb4pa9sq.amazonaws.com/photos/puppy.jpg?
rk=e2c69a31
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Fri, 12 Oct 2007 01:12:56 GMT
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>TemporaryRedirect</Code>
```

```
<Message>Please re-send this request to the specified temporary endpoint.  
Continue to use the original request endpoint for future requests.</Message>  
<Endpoint>awsexamplebucket1.s3-gztb4pa9sq.amazonaws.com</Endpoint>  
</Error>
```

Temporäre Umleitungsantwort der SOAP API

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

```
<soapenv:Body>  
  <soapenv:Fault>  
    <Faultcode>soapenv:Client.TemporaryRedirect</Faultcode>  
    <Faultstring>Please re-send this request to the specified temporary endpoint.  
Continue to use the original request endpoint for future requests.</Faultstring>  
    <Detail>  
      <Bucket>images</Bucket>  
      <Endpoint>s3-gztb4pa9sq.amazonaws.com</Endpoint>  
    </Detail>  
  </soapenv:Fault>  
</soapenv:Body>
```

Überlegungen zu DNS

Eine der Entwurfsanforderungen von Amazon S3 ist eine extrem hohe Verfügbarkeit. Unter anderem erfüllen wir diese Anforderung, indem wir die dem Amazon-S3-Endpunkt in DNS zugeordnete IP-Adressen nach Bedarf aktualisieren. Diese Änderungen werden in kurzlebigen Clients automatisch reflektiert, nicht jedoch in einigen langlebigen Clients. Für langlebige Clients ist eine spezielle Maßnahme erforderlich, um den Amazon-S3-Endpunkt regelmäßig wieder aufzulösen, um von diesen Änderungen zu profitieren. Weitere Informationen zu virtuellen Maschinen (VMs) finden Sie hier:

- Für Java speichert die JVM von Sun die DNS-Suchen standardmäßig auf unbegrenzte Zeit. Informationen zum Ändern dieses Verhaltens finden Sie im Abschnitt „InetAddress Caching“ in der [InetAddress-Dokumentation](#).

- Für PHP speichert die persistente PHP VM, die in den gebräuchlichsten Bereitstellungskonfigurationen ausgeführt wird, DNS-Suchen, bis die VM neu gestartet wird. Weitere Informationen finden Sie in den [getHostByName-PHP-Dokumenten](#).

Behandlung von REST- und SOAP-Fehlern

Themen

- [Die REST-Fehlerantwort](#)
- [Die SOAP-Fehlerantwort](#)
- [Bewährte Methoden für Amazon-S3-Fehler](#)

Dieser Abschnitt beschreibt REST- und SOAP-Fehler und wie sie gehandhabt werden.

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

Die REST-Fehlerantwort

Wenn auf eine REST-Anfrage ein Fehler zurückgegeben wird, ist die HTTP-Antwort wie folgt aufgebaut:

- Ein XML-Fehlerdokument als Antworttext
- Content-Type: application/xml
- Der entsprechende 3xx-, 4xx- oder 5xx-HTTP-Statuscode

Nachfolgend finden Sie ein Beispiel für eine REST-Fehlermeldung.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The resource you requested does not exist</Message>
```



```
<Resource>/mybucket/myfoto.jpg</Resource>
<RequestId>4442587FB7D0A2F9</RequestId>
</Error>
```

Weitere Informationen zu Amazon-S3-Fehlern finden Sie unter [ErrorCodeList](#).

Antwort-Header

Die folgenden Antwort-Header werden von allen Operationen zurückgegeben:

- `x-amz-request-id`: Eine eindeutige ID, die das System jeder Anfrage zuordnet. In dem unwahrscheinlichen Fall, dass Sie Probleme mit Amazon S3 haben, kann Ihnen diese helfen, das Problem zu beheben.
- `x-amz-id-2`: Ein spezielles Token, das uns hilft, Probleme aufzulösen.

Antwort auf einen Fehler

Bei einer fehlerhaften Amazon-S3-Anforderung erhält der Client eine Fehlermeldung zurück. Das genaue Format der Fehlermeldung ist spezifisch für die API. Die REST-Fehlermeldung beispielsweise unterscheidet sich von der SOAP-Fehlermeldung. Alle Fehlermeldungen haben jedoch gemeinsame Elemente.

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

Fehlercode

Der Fehlercode ist eine Zeichenfolge, die eine Fehlerbedingung eindeutig identifiziert. Er ist für Programme vorgesehen, die Fehler dem Typ nach erkennen und verarbeiten. Viele Fehlercode gibt es für SOAP- und REST-APIs, einige sind jedoch spezifisch für die API. `NoSuchKey` beispielsweise ist universell, während `UnexpectedContent` nur in Reaktion auf eine ungültige REST-Anfrage auftreten kann. In jedem Fall haben SOAP-Fehlercodes ein Präfix, das in der Fehlercodetabelle angegeben ist, ein `NoSuchKey`-Fehler wird also in SOAP als `Client.NoSuchKey` zurückgegeben.

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

Fehlermeldung

Die Fehlermeldung enthält eine generische Beschreibung der Fehlerbedingung in englischer Sprache. Sie ist für dafür vorgesehen, von Menschen gelesen zu werden. Einfache Programme zeigen dem Endbenutzer die Meldung direkt an, wenn sie auf eine Fehlerbedingung treffen, die sie nicht kennen, oder die sie nicht verarbeiten können. Komplexere Programme mit einer umfangreicheren Fehlerverarbeitung und einer geeigneten Internationalisierung werden die Fehlermeldung eher ignorieren.

Weitere Details

Viele Fehlermeldungen enthalten zusätzliche strukturierte Daten, die von einem Entwickler gelesen und interpretiert werden, der Programmfehler diagnostiziert. Wenn Sie beispielsweise einen Content-MD5-Header mit einer REST PUT-Anfrage senden, die nicht mit dem auf dem Server berechneten Digest übereinstimmt, erhalten Sie einen BadDigest-Fehler. Die Fehlermeldung enthält auch als Details den von uns berechneten Digest, sowie den Digest, den Sie erwarten und uns mitgeteilt haben. Während der Entwicklung können Sie diese Informationen nutzen, um den Fehler zu diagnostizieren. In der Produktion könnte ein ordnungsgemäß funktionierendes Programm diese Informationen in seinem Fehlerprotokoll enthalten.

Die SOAP-Fehlerantwort

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

In SOAP wird eine Fehlermeldung als SOAP-Fehler an den Client zurückgegeben, mit dem HTTP-Antwortcode 500. Wenn Sie keinen SOAP-Fehler erhalten, war Ihre Anfrage erfolgreich. Der Amazon S3SOAP-Fehlercode besteht aus einem SOAP 1.1-Standardfehlercode („Server“ oder „Client“),

verknüpft mit dem Amazon-S3-spezifischen Fehlercode. Beispiel: "Server.InternalError" oder "Client.NoSuchBucket". Das SOAP-Fehlerzeichenfolgeelement enthält eine generische, vom Menschen lesbare Fehlermeldung in englischer Sprache. Das SOAP-Fehlerdetailelement schließlich enthält verschiedene für den Fehler relevante Informationen.

Wenn Sie beispielsweise versuchen, das nicht existierende Objekt "Fred" zu löschen, enthält der Rumpf der SOAP-Antwort den SOAP-Fehler "NoSuchKey".

Example

```
<soapenv:Body>
  <soapenv:Fault>
    <Faultcode>soapenv:Client.NoSuchKey</Faultcode>
    <Faultstring>The specified key does not exist.</Faultstring>
    <Detail>
      <Key>Fred</Key>
    </Detail>
  </soapenv:Fault>
</soapenv:Body>
```

Weitere Informationen zu Amazon-S3-Fehlern finden Sie unter [ErrorCodeList](#).

Bewährte Methoden für Amazon-S3-Fehler

Wenn Sie eine Anwendung für Amazon S3 entwerfen, sollten Sie darauf achten, dass Amazon-S3-Fehler auf geeignete Weise verarbeitet werden. Dieser Abschnitt beschreibt Aspekte, die Sie beim Entwurf Ihrer Anwendung berücksichtigen sollten.

Wiederholung bei InternalErrors

Interne Fehler sind Fehler, die innerhalb der Amazon-S3-Umgebung auftreten.

Anfragen, die eine InternalError-Antwort erhalten, wurden möglicherweise nicht verarbeitet. Gibt eine PUT-Anfrage beispielsweise einen InternalError zurück, ruft ein nachfolgendes GET möglicherweise den alten Wert oder den aktuellen Wert ab.

Wenn Amazon S3 eine InternalError-Antwort zurückgibt, wiederholen Sie die Anforderung.

Optimieren der Anwendung für wiederholte SlowDown-Fehler

Wie jedes verteilte System besitzt S3 Schutzmechanismen, die einen beabsichtigten oder nicht beabsichtigten übermäßigen Ressourcenverbrauch erkennen und entsprechend reagieren.

SlowDown-Fehler können auftreten, wenn eine hohe Anfragerate einen dieser Mechanismen auslöst. Eine Reduzierung Ihrer Anfragerate verringert oder eliminiert Fehler dieses Typs. Im Allgemeinen werden die meisten Benutzer diese Fehler nicht regelmäßig sehen. Wenn Sie jedoch weitere Informationen wünschen oder viele oder unerwartete SlowDown-Fehler sehen, posten Sie bitte in unserem [Amazon-S3-Entwicklerforum](#) oder melden Sie sich für AWS Support an <https://aws.amazon.com/premiumsupport/>.

Isolieren von Fehlern

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

Amazon S3 unterstützt verschiedene Fehlercodes, die sowohl von der SOAP, als auch von der REST-API verwendet werden. Die SOAP API gibt Amazon-S3-Standardfehlercodes zurück. Die REST-API ist darauf ausgelegt, wie ein HTTP-Standardserver auszusehen und mit vorhandenen HTTP-Clients zu interagieren (z. B. Browsern, HTTP-Clientbibliotheken, Proxies, Caches usw.). Um sicherzustellen, dass die HTTP-Clients Fehler ordnungsgemäß verarbeiten, bilden wir jeden Amazon-S3-Fehler auf einen HTTP-Statuscode ab.

HTTP-Statuscodes sind weniger ausdrucksstark als Amazon-S3-Fehlercodes und enthalten weniger Informationen über den Fehler. Beispielsweise werden die Amazon-S3-Fehler NoSuchKey und NoSuchBucket auf den Statuscode HTTP 404 Not Found abgebildet.

Obwohl HTTP-Statuscodes weniger Informationen über den Fehler enthalten, können Clients, die HTTP verstehen, aber nicht die Amazon-S3-API, den Fehler normalerweise korrekt verarbeiten.

Wenn Sie also Fehler verarbeiten oder Amazon-S3-Fehler an Endbenutzer melden, verwenden Sie den Amazon-S3-Fehlercode statt des HTTP-Statuscodes, weil dieser die meisten Informationen über den Fehler enthält. Beim Debuggen Ihrer Anwendung sollten Sie außerdem das lesbare Element <Details> der XML-Fehlerantwort heranziehen.

Referenz für Entwickler

Dieser Anhang umfasst folgende Abschnitte.

Themen

- [Anhang A: Verwenden der SOAP-API](#)
- [Anhang b: Authentifizieren von Anfragen \(AWS Signaturversion 2\)](#)

Anhang A: Verwenden der SOAP-API

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Anstatt SOAP zu verwenden, empfehlen wir, entweder die REST-API oder die AWS SDKs zu verwenden.

Dieser Abschnitt enthält spezifische Informationen für die Amazon S3 SOAP-API.

Note

SOAP-Anfragen, und zwar sowohl authentifizierte als auch anonyme, müssen mit SSL an Amazon S3 gesendet werden. Amazon S3 gibt einen Fehler zurück, wenn Sie eine SOAP-Anfrage über HTTP senden.

Themen

- [Allgemeine SOAP-API-Elemente](#)
- [Authentifizieren von SOAP-Anforderungen](#)
- [Festlegen der Zugriffsrichtlinie mit SOAP](#)

Allgemeine SOAP-API-Elemente

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

Sie können mit Amazon S3 unter Verwendung von SOAP 1.1 über HTTP interagieren. Die Amazon S3 WSDL, die die Amazon-S3-API maschinenlesbar beschreibt, ist unter folgender Adresse verfügbar: <https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>. Das Amazon-S3-Schema ist unter <https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.xsd> verfügbar.

Die meisten Benutzer interagieren mit Amazon S3 über ein SOAP-Toolkit, das auf ihre Sprache und Entwicklungsumgebung zugeschnittenen ist. Verschiedene Toolkits machen die Amazon-S3-API auf unterschiedliche Weise verfügbar. Bitte konsultieren Sie Ihre spezifische Toolkit-Dokumentation für seine Verwendung. Dieser Abschnitt veranschaulicht die Amazon-S3-SOAP-Operationen in einer Toolkit-unabhängigen Weise durch Darstellen der XML-Anfragen und Antworten, wie sie „online“ erscheinen.

Allgemeine Elemente

Sie können die folgenden autorisierungsbezogenen Elemente in jede SOAP-Anfrage aufnehmen:

- **AWSAccessKeyId**: Die AWS-Zugriffsschlüssel-ID des Anforderers
- **Timestamp**: Die aktuelle Systemzeit
- **Signature**: Die Signatur für die Anforderung

Authentifizieren von SOAP-Anforderungen

Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Wir empfehlen, statt SOAP entweder die REST-API oder die AWS-SDKs zu verwenden.

Jede nicht anonyme Anfrage muss Authentifizierungsinformationen enthalten, um die Identität des Prinzipals einzurichten, der die SOAP-Anfrage stellt. In SOAP wird die Authentifizierungsinformation in den folgenden Elemente der SOAP-Anfrage abgelegt:

- Ihre AWS-Zugriffsschlüssel-ID.

Note

Bei authentifizierten SOAP-Anfragen werden keine temporären Sicherheitsanmeldeinformationen unterstützt. Weitere Informationen den verschiedenen Anmeldeinformationen finden Sie unter [Senden von Anforderungen](#).

- **Timestamp:** Dabei muss es sich um ein dateTime (siehe <http://www.w3.org/TR/xmlschema-2/#dateTime>) in der Zeitzone Coordinated Universal Time (Greenwich Mean Time) handeln, beispielsweise 2009-01-01T12:00:00.000Z. Die Autorisierung schlägt fehl, wenn dieser Zeitstempel um mehr als 15 Minuten von der Uhr auf Amazon-S3-Servern abweicht.
- **Signature:** Der RFC 2104 HMAC-SHA1-Digest (siehe <http://www.ietf.org/rfc/rfc2104.txt>) der Verkettung von "AmazonS3" + OPERATION + Zeitstempel, wobei Ihr geheimer AWS-Zugriffsschlüssel als Schlüssel verwendet wird. In der folgenden CreateBucket-Beispielanfrage würde das Signatur-Element den HMAC-SHA1-Digest mit dem Wert "AmazonS3CreateBucket2009-01-01T12:00:00.000Z" enthalten:

In der folgenden CreateBucket-Beispielanfrage würde das Signatur-Element den HMAC-SHA1-Digest mit dem Wert "AmazonS3CreateBucket2009-01-01T12:00:00.000Z" enthalten:

Example

```
<CreateBucket xmlns="https://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Acl>private</Acl>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2009-01-01T12:00:00.000Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</CreateBucket>
```

Note

SOAP-Anfragen, und zwar sowohl authentifizierte als auch anonyme, müssen mit SSL an Amazon S3 gesendet werden. Amazon S3 gibt einen Fehler zurück, wenn Sie eine SOAP-Anfrage über HTTP senden.

⚠ Important

Aufgrund von unterschiedlichen Interpretationen dahingehend, wie eine zusätzliche Zeitgenauigkeit wegfallen kann, sollten .NET-Benutzer darauf achten, Amazon S3 keine übermäßig spezifischen Zeitstempel zu senden. Dies kann bewerkstelligt werden, indem manuell `DateTime`-Objekte mit einer Millisekunde Genauigkeit erstellt werden.

Festlegen der Zugriffsrichtlinie mit SOAP

ℹ Note

Die SOAP-Unterstützung über HTTP ist veraltet, SOAP steht über HTTPS aber noch zur Verfügung. Neue Amazon-S3-Funktionen werden unter SOAP nicht unterstützt. Anstatt SOAP zu verwenden, empfehlen wir, entweder die REST-API oder die AWS SDKs zu verwenden.

Die Zugriffskontrolle kann beim Schreiben eines Buckets oder Objekts festgelegt werden, indem das Element "AccessControlList" in die Anforderung an `CreateBucketPutObjectInline`, oder `PutObject` aufgenommen wird. Das `AccessControlList` Element wird unter [beschrieben Identity and Access Management für Amazon S3](#). Wenn bei diesen Operationen keine Zugriffskontrollliste angegeben wird, wird die Ressource mit einer Standardzugriffsrichtlinie erstellt, die dem Anforderer `FULL_CONTROL`-Zugriff gewährt (dies ist auch dann der Fall, wenn es sich bei der Anfrage um eine `PutObject` Oder-Anfrage für ein `PutObjectInline` Objekt handelt, das bereits existiert).

Nachfolgend sehen Sie eine Anfrage, die Daten in ein Objekt schreibt, das Objekt lesbar für anonyme Prinzipale macht, und dem angegebenen Benutzer `FULL_CONTROL`-Rechte für den Bucket erteilt (die meisten Entwickler wollen sich selbst `FULL_CONTROL`-Zugriff auf ihren eigenen Bucket geben).

Example

Die nachfolgende Anfrage schreibt Daten in ein Objekt und macht das Objekt lesbar für anonyme Prinzipale.

Sample Request

```
<PutObjectInline xmlns="https://doc.s3.amazonaws.com/2006-03-01">
```



```

<Bucket>quotes</Bucket>
<Key>Nelson</Key>
<Metadata>
  <Name>Content-Type</Name>
  <Value>text/plain</Value>
</Metadata>
<Data>aGEtaGE=</Data>
<ContentLength>5</ContentLength>
<AccessControlList>
  <Grant>
    <Grantee xsi:type="CanonicalUser">
      <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>chriscustomer</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
  <Grant>
    <Grantee xsi:type="Group">
      <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
    </Grantee>
    <Permission>READ</Permission>
  </Grant>
</AccessControlList>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2009-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</PutObjectInline>

```

Sample Response

```

<PutObjectInlineResponse xmlns="https://s3.amazonaws.com/doc/2006-03-01">
  <PutObjectInlineResponse>
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;</ETag>
    <LastModified>2009-01-01T12:00:00.000Z</LastModified>
  </PutObjectInlineResponse>
</PutObjectInlineResponse>

```

Die Zugriffskontrollrichtlinie kann für einen vorhandenen Bucket oder ein vorhandenes Objekt mit den Methoden `GetBucketAccessControlPolicy`, `GetObjectAccessControlPolicy`, `SetBucketAccessControlPolicy` und `SetObjectAccessControlPolicy` gelesen oder eingerichtet werden. Weitere Informationen finden Sie in der detaillierten Erklärung dieser Methoden.

Anhang b: Authentifizieren von Anfragen (AWS Signaturversion 2)

Important

In diesem Abschnitt wird beschrieben, wie Anfragen mit AWS Signature Version 2 authentifiziert werden. Signature Version 2 wird deaktiviert (veraltet), Amazon S3 akzeptiert nur API-Anforderungen, die mit Signature Version 4 signiert wurden. Weitere Informationen finden Sie unter [AWS Signature Version 2 wurde für Amazon S3 deaktiviert \(veraltet\)](#). Signature Version 4 wird in allen AWS-Regionen Versionen unterstützt und ist die einzige Version, die für neue Regionen unterstützt wird. Weitere Informationen finden Sie unter [Authentifizieren von Anfragen \(AWS Signature Version 4\)](#) in der Amazon Simple Storage Service API-Referenz.

Amazon S3 ermöglicht das Ermitteln der zum Signieren einer Anforderung verwendeten API Signature Version. Um Beeinträchtigungen der betrieblichen Prozesse zu vermeiden, muss festgestellt werden, ob Workflows mit Signature Version 2 signieren. Diese Workflows müssen dann auf die Verwendung von Signature Version 4 umgestellt werden.

- Wenn Sie CloudTrail Ereignisprotokolle verwenden (empfohlene Option), erfahren [Identifizieren von Amazon S3 Signature Version 2-Anfragen mithilfe von CloudTrail](#) Sie hier, wie Sie solche Anfragen abfragen und identifizieren können.
- Wenn Sie die Amazon S3 Server Access-Protokolle verwenden, vgl. [Identifizieren von Signature-Version-2-Anforderungen mittels Amazon-S3-Zugriffsprotokollen](#)

Themen

- [Authentifizieren von Anforderungen mit der REST-API](#)
- [Anfragen signieren und authentifizieren REST](#)
- [Browserbasierte Uploads mit POST \(AWS Signaturversion 2\)](#)

Authentifizieren von Anforderungen mit der REST-API

Beim Zugriff auf Amazon S3 mit REST müssen Sie in Ihrer Anfrage die folgenden Elemente angeben, damit diese authentifiziert werden kann.:

Anfordern von Elementen

- AWS Zugriffsschlüssel-ID — Jede Anfrage muss die Zugriffsschlüssel-ID der Identität enthalten, die Sie zum Senden Ihrer Anfrage verwenden.
- Signatur – Jede Anforderung muss eine gültige Anforderungssignatur enthalten, oder die Anforderung wird abgelehnt.

Eine Anforderungssignatur wird unter Verwendung Ihres geheimen Zugriffsschlüssels berechnet, das ist ein gemeinsames Geheimnis, das nur Sie und AWS kennen.

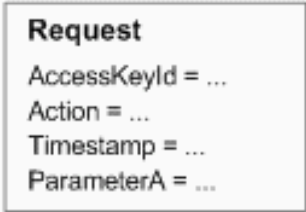
- Zeitstempel – Jede Anfrage muss Datum und Uhrzeit des Zeitpunktes enthalten, zu dem die Anfrage erstellt wurde, dargestellt als Zeichenfolge in UTC.
- Datum – Jede Anfrage muss den Zeitstempel der Anfrage enthalten.

Abhängig von der verwendeten API-Aktion können Sie ein Ablaufdatum und eine Ablaufzeit für die Anfrage angeben, statt dem Zeitstempel oder zusätzlich zu diesem. Um festzustellen, was notwendig ist, lesen Sie im Authentifizierungsthema für die jeweiligen Aktion nach.

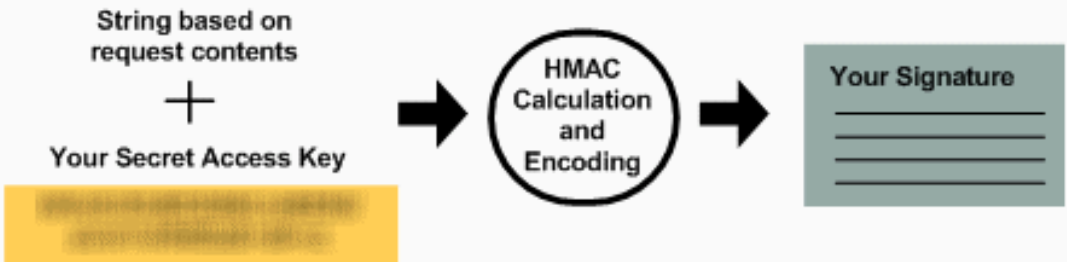
Nachfolgend finden Sie die allgemeinen Schritte für die Authentifizierung von Anfragen für Amazon S3. Es wird vorausgesetzt, dass Sie über die erforderlichen Sicherheitsanmeldeinformationen, die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel verfügen.

You

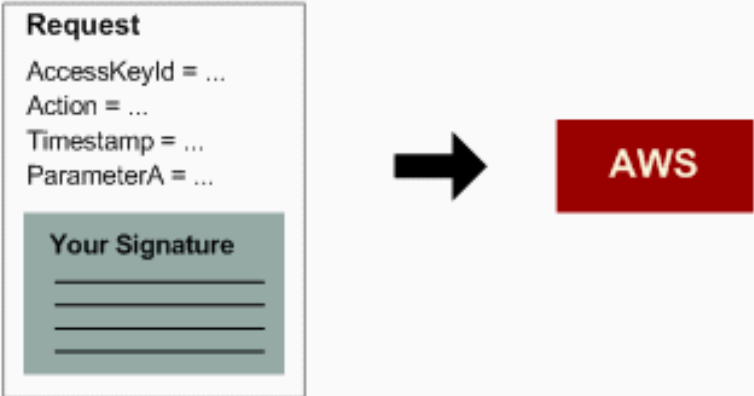
1 Create a request:



2 Create an HMAC-SHA1 signature:



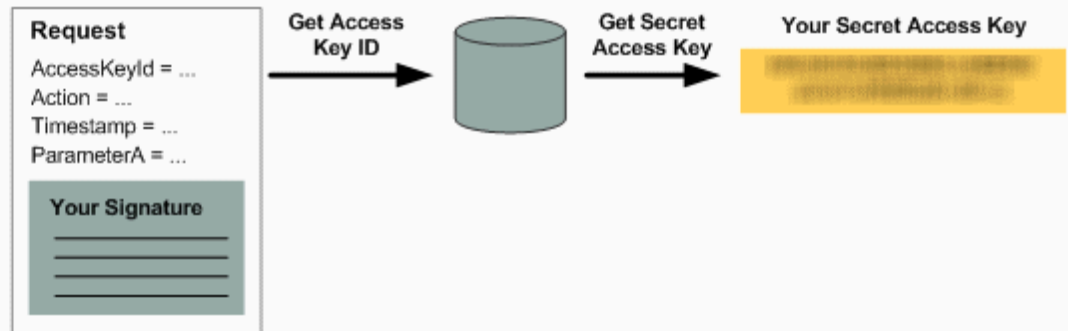
3 Send the request and signature to AWS:



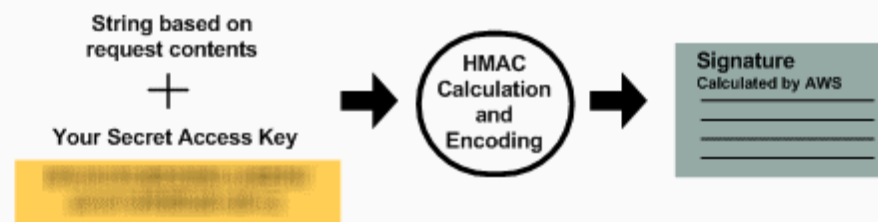
1	Konstruieren Sie eine Anfrage an AWS
2	Berechnung der Signatur unter Verwendung Ihres geheimen Zugriffsschlüssels.
3	Senden Sie die Anfrage an Amazon S3. Nehmen Sie Ihre Zugriffsschlüssel-ID und die Signatur in Ihre Anforderung auf. Amazon S3 führt die nächsten drei Schritte aus.

AWS

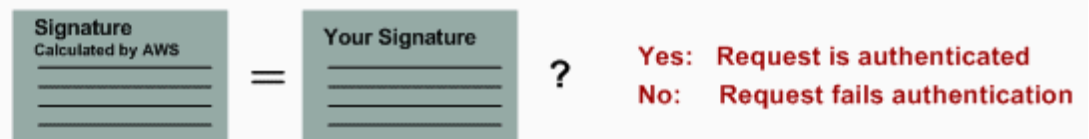
4 Retrieve your Secret Access Key:



5 Create an HMAC-SHA1 signature:



6 Compare the two signatures:



4 Amazon S3 verwendet die Zugriffsschlüssel-ID, um Ihren geheimen Zugriffsschlüssel abzurufen.

5 Amazon S3 berechnet eine Signatur aus den Anforderungsdaten und dem geheimen Zugriffsschlüssel mit demselben Algorithmus, mit dem die in der Anforderung gesendete Signatur berechnet wurde.

6 Wenn die von Amazon S3 generierte Signatur mit der in der Anforderung gesendete n Signatur übereinstimmt, wird die Anforderung als authentisch eingestuft. Falls der Vergleich fehlschlägt, wird die Anfrage verworfen, und Amazon S3 gibt eine Fehlerantwort zurück.

Detaillierte Authentifizierungsdaten

Weitere Informationen zur REST-Authentifizierung finden Sie unter [Anfragen signieren und authentifizieren REST](#).

Anfragen signieren und authentifizieren REST

Themen


- [Verwenden von temporären Sicherheitsanmeldeinformationen](#)
- [Der Authentifizierungsheader](#)
- [Anfordern einer Kanonisierung für die Signatur](#)
- [Das Element konstruieren CanonicalizedResource](#)
- [Konstruieren des Elements CanonicalizedAmzHeaders](#)
- [Positionelle und benannte Header-Elemente HTTP StringToSign](#)
- [Zeitstempel-Anforderung](#)
- [Authentifizierungsbeispiele](#)
- [RESTProbleme beim Signieren von Anfragen](#)
- [Alternative zur Authentifizierung der Abfragezeichenfolge](#)

Note

Dieses Thema erklärt Authentifizierungsanfragen unter Verwendung von Signature Version 2. Amazon S3 unterstützt jetzt die neueste Signature Version 4. Diese neueste Signatur-Version wird in allen Regionen unterstützt. Alle neuen Regionen unterstützen nach dem 30. Januar 2014 nur Signature Version 4. Weitere Informationen finden Sie unter [Authentifizieren von Anfragen \(AWS Signature Version 4\)](#) in der Amazon Simple Storage API Service-Referenz.

Die Authentifizierung ist der Prozess, Ihre Identität gegenüber dem System nachzuweisen. Die Identität ist ein wichtiger Faktor in den Zugriffssteuerungsentscheidungen von Amazon S3. Anforderungen werden zum Teil basierend auf der Identität des Auftraggebers zugelassen oder abgewiesen. Das Recht, Buckets zu erstellen, ist beispielsweise für registrierte Entwickler reserviert (standardmäßig), und das Recht, Objekte in einem Bucket zu erstellen, ist für den Eigentümer des betreffenden Buckets reserviert. Als Entwickler machen Sie Anforderungen, für die diese

Berechtigungen gelten müssen, deshalb müssen Sie Ihre Identität gegenüber dem System belegen, indem Sie Ihre Anforderungen authentifizieren. In diesem Abschnitt erfahren Sie mehr darüber.

 Note

Der Inhalt dieses Abschnitts gilt nicht für. HTTP POST Weitere Informationen finden Sie unter [Browserbasierte Uploads mit POST \(AWS Signaturversion 2\)](#).

Amazon S3 REST API verwendet ein benutzerdefiniertes HTTP Schema, das auf einem Schlüssel HMAC (Hash Message Authentication Code) für die Authentifizierung basiert. Um eine Anforderung zu authentifizieren, verknüpfen Sie zuerst ausgewählte Elemente der Anforderung, um eine Zeichenfolge zu erstellen. Anschließend verwenden Sie Ihren AWS geheimen Zugriffsschlüssel, um den Wert HMAC dieser Zeichenfolge zu berechnen. Informell nennen wir diesen Vorgang „Signieren der Anfrage“, und wir nennen die Ausgabe des HMAC Algorithmus die Signatur, weil sie die Sicherheitseigenschaften einer echten Signatur simuliert. Schließlich fügen Sie diese Signatur als Parameter der Anforderung hinzu. Dazu verwenden Sie die in diesem Abschnitt beschriebene Syntax.

Wenn das System eine authentifizierte Anfrage empfängt, ruft es den AWS geheimen Zugriffsschlüssel ab, den Sie angeblich haben, und verwendet ihn auf die gleiche Weise, um eine Signatur für die empfangene Nachricht zu berechnen. Anschließend vergleicht es die berechnete Signatur mit der Signatur, die der Anforderer vorgezeigt hat. Wenn die beiden Signaturen übereinstimmen, kommt das System zu dem Schluss, dass der Anforderer Zugriff auf den AWS geheimen Zugriffsschlüssel haben muss, und handelt daher mit der Autorität des Prinzipals, dem der Schlüssel ausgestellt wurde. Stimmen die beiden Signaturen nicht überein, wird die Anforderung verworfen und das System antwortet mit einer Fehlermeldung.

Example Authentifizierte Amazon S3 S3-Anfrage REST

```
GET /photos/puppy.jpg HTTP/1.1
Host: awsexamplebucket1.us-west-1.s3.amazonaws.com
Date: Tue, 27 Mar 2007 19:36:42 +0000
```

```
Authorization: AWS AKIAIOSFODNN7EXAMPLE:
qgk2+6Sv9/oM7G3qLEjTH1a1l1g=
```

Verwenden von temporären Sicherheitsanmeldeinformationen

Wenn Sie Ihre Anfrage unter Verwendung temporärer Sicherheitsanmeldeinformationen signieren (siehe [Senden von Anforderungen](#)), müssen Sie das entsprechende Sicherheitstoken in Ihre Anfrage aufnehmen, indem Sie den `x-amz-security-token`-Header hinzufügen.

Wenn Sie mithilfe von temporäre Sicherheitsanmeldedaten abrufen AWS Security Token Service API, enthält die Antwort temporäre Sicherheitsanmeldedaten und ein Sitzungstoken. Sie geben den Wert des Sitzungstokens im `x-amz-security-token`-Header an, wenn Sie Anfragen an Amazon S3 senden. Weitere Informationen zu den AWS Security Token Service API bereitgestellten Funktionen IAM finden Sie im AWS Security Token Service APIReferenzhandbuch unter [Aktion](#).

Der Authentifizierungsheader

Amazon S3 REST API verwendet den HTTP `Authorization` Standard-Header, um Authentifizierungsinformationen zu übergeben. (Der Name des Standardheaders ist unglücklich gewählt, weil er eine Authentifizierung weitergibt, keine Autorisierung.) Unter dem Amazon-S3-Authentifizierungsschema hat der Autorisierungsheader die folgende Form:

```
Authorization: AWS AWSAccessKeyId:Signature
```

Entwickler erhalten bei der Registrierung eine AWS Zugriffsschlüssel-ID und einen AWS geheimen Zugriffsschlüssel. Für die Anforderungs-Authentifizierung identifiziert das `AWSAccessKeyId`-Element die Zugriffsschlüssel-ID, die für die Berechnung der Signatur verwendet wurde, und indirekt für den Entwickler steht, der die Anforderung gestellt hat.

Bei dem `Signature` Element handelt es sich um das RFC 2104 HMAC — SHA1 der ausgewählten Elemente aus der Anfrage, sodass der `Signature` Teil des `Authorization`-Headers von Anfrage zu Anfrage variieren kann. Stimmt die vom System berechnete Anforderungssignatur mit der in der Anfrage `Signature` enthaltenen überein, hat der Antragsteller den Besitz des AWS geheimen Zugangsschlüssels nachgewiesen. die Anforderung wird unter der Identität verarbeitet, und mit der Genehmigung des Entwicklers, dem der Schlüssel ausgestellt wurde.

Die folgende Pseudogrammatik verdeutlicht den Aufbau des `Authorization`-Anforderungs-Headers. (In dem Beispiel steht `\n` für den Unicode-Code `U+000A`, häufig auch als Newline bezeichnet.)

```
Authorization = "AWS" + " " + AWSAccessKeyId + ":" + Signature;
```



```
Signature = Base64( HMAC-SHA1( UTF-8-Encoding-Of(YourSecretAccessKey), UTF-8-Encoding-Of( StringToSign ) ) );
```

```
StringToSign = HTTP-Verb + "\n" +  
Content-MD5 + "\n" +  
Content-Type + "\n" +  
Date + "\n" +  
CanonicalizedAmzHeaders +  
CanonicalizedResource;
```

```
CanonicalizedResource = [ "/" + Bucket ] +  
<HTTP-Request-URI, from the protocol name up to the query string> +  
[ subresource, if present. For example "?acl", "?location", or "?logging"];
```

```
CanonicalizedAmzHeaders = <described below>
```

HMAC- SHA1 ist ein durch [RFC2104 definierter Algorithmus — Keyed-Hashing](#) for Message Authentication. Der Algorithmus nimmt zwei Byte-Zeichenfolgen als Eingabe entgegen, einen Schlüssel und eine Nachricht. Verwenden Sie für die Amazon S3 S3-Anforderungsauthentifizierung Ihren AWS geheimen Zugriffsschlüssel (YourSecretAccessKey) als Schlüssel und die UTF -8-Kodierung von StringToSign als Nachricht. Die Ausgabe von HMAC - SHA1 ist ebenfalls eine Byte-Zeichenfolge, die als Digest bezeichnet wird. Der Signature-Anforderungsparameter wird durch eine Base64-Codierung dieses Digest erstellt.

Anfordern einer Kanonisierung für die Signatur

Wenn das System eine authentifizierte Anforderung erhält, vergleicht es die berechnete Anforderungssignatur mit der in der Anforderung im bereitgestellten Signatur, wie bereits beschriebene StringToSign. Aus diesem Grund müssen Sie die Signatur nach derselben Methode berechnen, die auch Amazon S3 verwendet. Wir bezeichnen diesen Prozess, bei dem eine Anforderung in die für die Signatur vereinbarte Form gebracht wird, als Kanonisierung.

Das Element konstruieren CanonicalizedResource

CanonicalizedResource stellt die für die Anfrage relevante Amazon-S3-Ressource dar. Konstruieren Sie es für eine REST Anfrage wie folgt:

Starten des Prozesses

- 1 Beginnen Sie mit einer leeren Zeichenfolge ("").

- 2 Wenn die Anfrage einen Bucket unter Verwendung des HTTP Host-Headers spezifiziert (virtuelles Hosting-Stil), fügen Sie den Bucket-Namen mit einem vorangestellten Namen an "/" (z. B. „/bucketname“). Für Anforderungen im Pfad-Stil und Anforderungen, die an keinen spezifischen Bucket gerichtet sind, machen Sie nichts. Weitere Informationen zum Wiederholen von Anforderungen finden Sie unter [Virtuelles Hosting bei Buckets](#).

Für eine Anfrage im virtuellen Hosting-Stil "puppy.jpg" ist dies „/awsexamplebucket1“. `https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/CanonicalizedResource`

Für die Anfrage im Pfadstil, "photos/puppy.jpg", ist das „“. `https://s3.us-west-1.amazonaws.com/awsexamplebucket1/CanonicalizedResource`

- 3 Hängen Sie den Pfadteil der nicht dekodierten HTTP Anfrage an, bis zuURI, aber ohne die Abfragezeichenfolge.

Für eine Anfrage im virtuellen Hosting-Stil "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg" ist das "/awsexamplebucket1/photos/puppy.jpg". `CanonicalizedResource`

Für eine Anfrage im Pfadstil, "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", ist das "/awsexamplebucket1/photos/puppy.jpg". An dieser Stelle ist der `CanonicalizedResource` für Anforderungen mit virtuell gehostetem Stil und Pfad-Stil gleich.

Bei einer Anfrage, die sich nicht auf einen Bucket bezieht, wie z. B. [GETService](#), fügen Sie „/“ an.

4 Wenn sich die Anforderung an eine Subressource richtet, wie beispielsweise `?versioning`, `?location`, `?acl`, `?lifecycle`, oder `?versionid`, fügen Sie die Subressource an, gegebenenfalls ihren Wert und das Fragezeichen. Beachten Sie, dass bei mehreren Unterressourcen die Unterressourcen lexikographisch nach dem Namen der Unterressource sortiert und durch '&' getrennt werden müssen, z. B. `?acl&=versionIdvalue`.

Die Unterressourcen, die bei der Erstellung des CanonicalizedResource Elements berücksichtigt werden müssen, sind ACL, Lebenszyklus, Speicherort, Protokollierung, Benachrichtigung, Richtlinie, `partNumber`, `UploadsrequestPayment`, `uploadId`, `VersionierungversionId`, Versionen und Website.

Wenn die Anfrage Abfragezeichenfolgen-Parameter angibt, die die Antwort-Header-Werte überschreiben (siehe [Get Object](#)), fügen Sie die Abfragezeichenfolgen-Parameter und ihre Werte an. Bei einer Signatur codieren Sie diese Werte nicht. Bei einer Anforderung dagegen müssen Sie diese Parameterwerte codieren. Zu den Parametern der Abfragezeichenfolge in einer GET Anfrage gehören `response-content-type`, `response-content-language`, `response-expires` und `response-cache-control` `response-content-disposition` `response-content-encoding`

Der `delete` Abfragezeichenfolgenparameter muss enthalten sein, wenn Sie die Löschanforderung CanonicalizedResource für mehrere Objekte erstellen.

Elemente von CanonicalizedResource, die aus der HTTP Anfrage stammen, URI sollten wörtlich so signiert werden, wie sie in der HTTP Anforderung vorkommen, einschließlich der URL -Kodierung von Metazeichen.

Sie CanonicalizedResource könnten sich von der HTTP Anfrage-URI unterscheiden. Insbesondere, wenn Ihre Anfrage den HTTP Host Header verwendet, um einen Bucket zu spezifizieren, erscheint der Bucket nicht im HTTP Request-URI. Die CanonicalizedResource beinhaltet den Bucket jedoch weiterhin. Parameter der Abfragezeichenfolge können auch in der Anfrage vorkommenURI, sind aber nicht darin enthaltenCanonicalizedResource. Weitere Informationen finden Sie unter [Virtuelles Hosting bei Buckets](#).

Konstruieren des Elements CanonicalizedAmzHeaders

Um den CanonicalizedAmzHeaders Teil von zu erstellenStringToSign, wählen Sie alle HTTP Anforderungsheader aus, die mit 'x-amz-' beginnen (wobei die Groß- und Kleinschreibung nicht berücksichtigt wird), und gehen Sie wie folgt vor.

CanonicalizedAmzHeaders verarbeiten

- 1 Wandelt jeden HTTP Header-Namen in Kleinbuchstaben um. Beispielsweise wird 'X-Amz-Date' zu 'x-amz-date'.
- 2 Sortieren Sie die Header alphabetisch nach dem Headernamen.
- 3 Kombinieren Sie Header-Felder mit demselben Namen zu einem „Header-Name:comma-separated-value-list“-Paar, wie in RFC 2616, Abschnitt 4.2 vorgeschrieben, ohne Leerzeichen zwischen den Werten. Die beiden Metadaten-Header 'x-amz-meta-username: fred' und 'x-amz-meta-username: barney' würden beispielsweise zu einem einzigen Header 'x-amz-meta-username: fred,barney' zusammengefasst.
- 4 Entfaltet lange Kopfzeilen, die sich über mehrere Zeilen erstrecken (wie in RFC 2616, Abschnitt 4.2 erlaubt), indem die Falzzeichen (einschließlich der neuen Zeile) durch ein einzelnes Leerzeichen ersetzt werden.
- 5 Löschen Sie alle Leerzeichen um den Doppelpunkt in der Kopfzeile. Beispielsweise würde der Header 'x-amz-meta-username: fred,barney' zu 'x-amz-meta-username:fred,barney'.
- 6 Schließlich fügen Sie jedem kanonisierten Header in der Ergebnisliste ein Neue-Zeile-Zeichen (U+000A) hinzu. Konstruieren Sie das CanonicalizedResource Element, indem Sie alle Header in dieser Liste zu einer einzigen Zeichenfolge verketteten.

Positionelle und benannte Header-Elemente HTTP StringToSign

Die ersten paar Header-Elemente von `StringToSign` (Content-Type, Date und Content-MD5) sind positioneller Natur. `StringToSign` enthält nicht die Namen dieser Header, sondern nur deren Werte aus der Anfrage. Im Gegensatz dazu sind die 'x-amz-'-Elemente benannt. Sowohl die Header-Namen als auch die Header-Werte erscheinen in `StringToSign`.

Wenn ein in der Definition von angeforderter Positionsheader in Ihrer Anfrage nicht vorhanden `StringToSign` ist (Content-Type oder Content-MD5 wenn er für Anfragen optional und für PUT Anfragen bedeutungslos ist), ersetzen Sie diese Position durch die leere Zeichenfolge („“). GET

Zeitstempel-Anforderung

Ein gültiger Zeitstempel (entweder unter Verwendung des HTTP Date Headers oder einer x-amz-date Alternative) ist für authentifizierte Anfragen erforderlich. Darüber hinaus muss der Client-Zeitstempel in einer authentifizierte Anfrage innerhalb eines Zeitraums von 15 Minuten zur Amazon-S3-Systemzeit liegen, wenn die Anfrage empfangen wird. Wenn dies nicht der Fall ist, schlägt die Anforderung mit RequestTimeTooSkewed-Fehlercode fehl. Diese Einschränkungen sollen verhindern, dass abgefangene Anforderungen böswillig wiederholt werden. Verwenden Sie für einen besseren Schutz vor Abhören den HTTPS Transport für authentifizierte Anfragen.

Note

Die Auswertungsbeschränkung im Hinblick auf das Anforderungsdatum gilt nur für authentifizierte Anforderungen, die keine Abfrage-String-Authentifizierung verwenden. Weitere Informationen finden Sie unter [Alternative zur Authentifizierung der Abfragezeichenfolge](#).

Einige HTTP Clientbibliotheken bieten nicht die Möglichkeit, den Date Header für eine Anfrage festzulegen. Wenn Sie Probleme damit haben, den Wert des 'Date'-Headers in kanonisierte Header aufzunehmen, können Sie den Zeitstempel für die Anforderung stattdessen auch mit einem 'x-amz-date'-Header einrichten. Der Wert des x-amz-date Headers muss in einem der RFC 2616 Formate vorliegen (<http://www.ietf.org/rfc/rfc2616.txt>). Wenn in einer Anforderung ein x-amz-date-Header vorhanden ist, ignoriert das System bei der Berechnung der Anforderungssignatur jeden Date-Header. Wenn Sie also den x-amz-date-Header aufnehmen, verwenden Sie die leere Zeichenfolge für Date, wenn Sie den StringToSign erstellen. Ein Beispiel finden Sie im nächsten Abschnitt.

Authentifizierungsbeispiele

Die Beispiele in diesem Abschnitt verwenden die (nicht funktionierenden) Anmeldeinformationen aus der folgenden Tabelle.

Parameter	Wert
AWSAccessKeyId	AKIAIOSFODNN7EXAMPLE
AWSSecretAccessKey	wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

In den `StringToSign`s des Beispiels ist die Formatierung nicht relevant, und `\n` steht für den Unicode-Punkt `U+000A`, allgemein als Neue-Zeile-Zeichen bezeichnet. Außerdem verwenden die Beispiele `" +0000"`, um die Zeitzone anzugeben. Sie können stattdessen `"GMT"` verwenden, um die Zeitzone anzugeben, aber die in den Beispielen gezeigten Signaturen werden unterschiedlich sein.

Objekt GET

In diesem Beispiel wird ein Objekt aus dem Bucket „`awsexamplebucket1`“ abgerufen.

Anforderung	StringToSign
<pre>GET /photos/puppy.jpg HTTP/1.1 Host: awsexamplebucket1.us- west-1.s3.amazonaws.com Date: Tue, 27 Mar 2007 19:36:42 +0000 Authorization: AWS AKIAIOSF0 DNN7EXAMPLE: qgk2+6Sv9/oM7G3qLEjTH1a1l1g=</pre>	<pre>GET\n \n \n Tue, 27 Mar 2007 19:36:42 +0000\n /awsexamplebucket1/photos/puppy.jpg</pre>

Beachten Sie, dass das den Bucket-Namen `CanonicalizedResource` enthält, das HTTP Request-URI nicht. (Der Bucket wird vom `Host-Header` spezifiziert.)

Note

Das folgende Python-Skript berechnet die vorhergehende Signatur unter Verwendung der bereitgestellten Parameter. Sie können dieses Skript verwenden, um Ihre eigenen Signaturen zu erstellen und die Schlüssel `StringToSign` gegebenenfalls zu ersetzen.

```
import base64
import hmac
from hashlib import sha1

access_key = 'AKIAIOSFODNN7EXAMPLE'.encode("UTF-8")
secret_key = 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'.encode("UTF-8")

string_to_sign = 'GET\n\n\nTue, 27 Mar 2007 19:36:42 +0000\n/awsexamplebucket1/
photos/puppy.jpg'.encode("UTF-8")
```

```
signature = base64.b64encode(
    hmac.new(
        secret_key, string_to_sign, sha1
    ).digest()
).strip()

print(f"AWS {access_key.decode()}:{signature.decode()}")
```

Objekt PUT

In diesem Beispiel wird ein Objekt in den Bucket „awsexamplebucket1“ eingefügt.

Anforderung	StringToSign
<pre>PUT /photos/puppy.jpg HTTP/1.1 Content-Type: image/jpeg Content-Length: 94328 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 21:15:45 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE: iqRzw+ileNPu1fhspnRs8n0jjIA=</pre>	<pre>PUT\n \n image/jpeg\n Tue, 27 Mar 2007 21:15:45 +0000\n /awsexamplebucket1/photos/puppy.jpg</pre>

Beachten Sie den Content-Type-Header in der Anfrage und im. StringToSign Beachten Sie auch, dass der Inhalt- in der leer gelassen MD5 wird StringToSign, da er in der Anfrage nicht vorhanden ist.

Auflisten

In diesem Beispiel wird der Inhalt des Buckets „awsexamplebucket1“ aufgelistet.

Anforderung	StringToSign
<pre>GET /?prefix=photos&max-keys=50&marker=puppy HTTP/1.1</pre>	<pre>GET\n \n</pre>

Anforderung	StringToSign
<pre>User-Agent: Mozilla/5.0 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 19:42:41 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE: m0WP8eCtspQl5Ahe6L1SozdX9YA=</pre>	<pre>\n Tue, 27 Mar 2007 19:42:41 +0000\n /awsexamplebucket1/</pre>

Beachten Sie den abschließenden Schrägstrich bei CanonicalizedResource und das Fehlen von Abfragezeichenfolgenparametern.

Fetch

In diesem Beispiel wird die Zugriffskontrollrichtlinien-Subressource für den Bucket „awsexamplebucket1“ abgerufen.

Anforderung	StringToSign
<pre>GET /?acl HTTP/1.1 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 19:44:46 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE: 82ZHIFiJc+WbcwFKGUVEQspPn+0=</pre>	<pre>GET\n \n \n Tue, 27 Mar 2007 19:44:46 +0000\n /awsexamplebucket1/?acl</pre>

Beachten Sie, dass der Parameter für die Subressource-Abfragezeichenfolge in der enthalten ist. CanonicalizedResource

Löschen

In diesem Beispiel wird ein Objekt über die path-style- und Date-Alternative aus dem Bucket „awsexamplebucket1“ entfernt.

Anforderung	StringToSign
<pre>DELETE /awsexamplebucket1/photos/puppy.jpg HTTP/1.1 User-Agent: dotnet Host: s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 21:20:27 +0000 x-amz-date: Tue, 27 Mar 2007 21:20:26 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE:XbyTlbQdu9Xw5o8P4iMwPktxQd8=</pre>	<pre>DELETE\n \n \n Tue, 27 Mar 2007 21:20:26 +0000\n /awsexamplebucket1/photos/puppy.jpg</pre>

Beachten Sie, dass wir die alternative Methode 'x-amz-date' zur Angabe des Datums verwendet haben (weil unsere Clientbibliothek uns beispielsweise daran gehindert hat, das Datum festzulegen). In diesem Fall hat x-amz-date Vorrang vor dem Date-Header. Aus diesem Grund muss der Datumseintrag in der Signatur den Wert des x-amz-date-Headers enthalten.

Hochladen

In diesem Beispiel wird ein Objekt mit Metadaten in einen virtuellen gehosteten Bucket CNAME im Stil eines Stils hochgeladen.

Anforderung	StringToSign
<pre>PUT /db-backup.dat.gz HTTP/1.1 User-Agent: curl/7.15.5 Host: static.example.com:8080 Date: Tue, 27 Mar 2007 21:06:08 +0000 x-amz-acl: public-read content-type: application/x-download Content-MD5: 4gJE4saaMU4BqNR0kLY+lw== X-Amz-Meta-ReviewedBy: joe@example.com X-Amz-Meta-ReviewedBy: jane@example.com X-Amz-Meta-FileChecksum: 0x02661779 X-Amz-Meta-ChecksumAlgorithm: crc32</pre>	<pre>PUT\n 4gJE4saaMU4BqNR0kLY+lw==\n application/x-download\n Tue, 27 Mar 2007 21:06:08 +0000\n x-amz-acl:public-read\n x-amz-meta-checksumalgorithm:crc32\n x-amz-meta-filechecksum:0x02661779\n x-amz-meta-reviewedby:joe@example.com,jane@example.com\n</pre>

Anforderung	StringToSign
<pre>Content-Disposition: attachment; filename=database.dat Content-Encoding: gzip Content-Length: 5913339 Authorization: AWS AKIAIOSFODNN7EXAMPLE LE: jtBQa0Aq+DkULFI8qrpwIjGEx0E=</pre>	<pre>/static.example.com/db-backup.dat .gz</pre>

Beachten Sie, wie die „x-amz“-Header sortiert, von Leerzeichen befreit und in Kleinbuchstaben umgewandelt wurden. Beachten Sie außerdem, dass mehrere Header mit demselben Namen unter Verwendung von Kommas zum Trennen der Werte verknüpft wurden.

Beachten Sie, dass nur die Entity-Header Content-Type und die Content-MD5 HTTP Entity-Header in der erscheinen. StringToSign Die anderen Content-* -Header erscheinen nicht.

Beachten Sie auch hier, dass der den Bucket-Namen CanonicalizedResource enthält, der HTTP Request- jedoch URI nicht. (Der Bucket wird vom Host-Header spezifiziert.)

Auflisten aller meiner Buckets

Anforderung	StringToSign
<pre>GET / HTTP/1.1 Host: s3.us-west-1.amazonaws.com Date: Wed, 28 Mar 2007 01:29:59 +0000 Authorization: AWS AKIAIOSFODNN7EXAMPLE:qGdzdE RIC03wnaRNKh60qZehG9s=</pre>	<pre>GET\n \n \n Wed, 28 Mar 2007 01:29:59 +0000\n /</pre>

Unicode-Schlüssel

Anforderung	StringToSign
<pre>GET /dictionary/fran%C3%A7ais/pr %c3%a9f%c3%a8re HTTP/1.1</pre>	<pre>GET\n \n</pre>

Anforderung	StringToSign
Host: s3.us-west-1.amazonaws.com Date: Wed, 28 Mar 2007 01:49:49 +0000 <i>Authorization: AWS AKIAIOSFODNN7EXAMPLE:DNEZGsoieTZ92F3bUfSPQcbGmLM=</i>	\n Wed, 28 Mar 2007 01:49:49 +0000\n /dictionary/fran%C3%A7ais/pr %c3%a9f%c3%a8re

Note

Die darin enthaltenen Elemente `StringToSign`, die aus der Anfrage- abgeleitet wurden, URI werden wörtlich genommen, einschließlich URL -Kodierung und Großschreibung.

RESTProbleme beim Signieren von Anfragen

Wenn die REST Anforderungsauthentifizierung fehlschlägt, beantwortet das System die Anfrage mit einem XML Fehlerdokument. Die in diesem Fehlerdokument enthaltene Information soll Entwicklern helfen, das Problem zu diagnostizieren. Insbesondere erkennen Sie an dem `StringToSign`-Element des `SignatureDoesNotMatch`-Fehlerdokuments genau, welche Anforderungs-Kanonisierung das System verwendet.

Einige Toolkits fügen automatisch Header ein, von denen Sie vorher nichts wussten, wie z. B. das Hinzufügen der Kopfzeile `Content-Type` während eines `PUT`. In den meisten dieser Fälle bleibt der Wert des eingefügten Headers konstant, sodass Sie fehlende Header unter Verwendung von Tools wie `Ethereal` oder `tcpmon` erkennen können.

Alternative zur Authentifizierung der Abfragezeichenfolge

Sie können bestimmte Arten von Anfragen authentifizieren, indem Sie die erforderlichen Informationen als Abfragezeichenfolgenparameter übergeben, anstatt den Header zu verwenden. `Authorization HTTP` Dies ist praktisch, um direkten Zugriff durch einen Browser von Dritten auf Ihre privaten Amazon-S3-Daten zu ermöglichen, ohne dass die Anfrage über einen Proxy gesendet wird. Die Idee ist, eine „vorsignierte“ Anfrage zu erstellen und sie so zu codieren, URL dass der Browser eines Endbenutzers sie abrufen kann. Darüber hinaus können Sie eine vorsignierte Anforderung durch die Angabe einer Ablaufzeit begrenzen.

Weitere Informationen zur Verwendung von Abfrageparametern zur Authentifizierung von Anfragen finden Sie unter [Authentifizieren von Anfragen: Verwenden von Abfrageparametern \(AWS Signature](#)

[Version 4](#)) in der Amazon Simple Storage API Service-Referenz. Beispiele für die Verwendung von AWS SDKs to zur Generierung vorsignierter URLs Dateien finden Sie unter. [Objekte mit vorsignierten teilen URLs](#)

Erstellen einer Signatur

Im Folgenden finden Sie ein Beispiel für eine authentifizierte Amazon S3 REST S3-Anfrage mit einer Abfragezeichenfolge.

```
GET /photos/puppy.jpg
?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=1141889120&Signature=vjbyPxybdZaNmGa
%2ByT272YEAiv4%3D HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.amazonaws.com
Date: Mon, 26 Mar 2007 19:37:58 +0000
```

Die Authentifizierungsmethode für die Anfrage mit einer Abfragezeichenfolge erfordert keine speziellen HTTP Header. Stattdessen werden die erforderlichen Authentifizierungselemente als Abfragezeichenfolgenparameter angegeben:

Abfrageze ichenfolgen- Parametername	Beispielwert	Beschreibung
AWSAccess KeyId	AKIAIOSFODNN7EXAMPLE	Ihre AWS Zugriffsschlüssel-ID. Gibt den AWS geheimen Zugriffsschlüssel an, mit dem die Anfrage signiert wurde, und indirekt die Identität des Entwicklers, der die Anfrage gestellt hat.
Expires	1141889120	Der Zeitpunkt, zu dem die Signatur abläuft, angegeben als Anzahl der Sekunden seit der Epoche (00:00:00 Uhr UTC am 1. Januar 1970). Eine Anforderung, die nach dieser Zeit empfangen wird (abhängig vom Server), wird abgelehnt.

Abfragezeichenfolgen-Parametername	Beispielwert	Beschreibung
Signature	vjbyPxybdZaNmGa%2B yT272YEAiv4%3D	Die URL Kodierung der Base64-Kodierung von - von. HMAC SHA1 StringToSign

Die Methode zur Authentifizierung der Abfragezeichenfolge unterscheidet sich leicht von der üblichen Methode, aber nur im Format des Signature-Anforderungsparameters und des StringToSign-Elements. Die folgende Pseudogrammatik verdeutlicht die Authentifizierungsmethode für die Abfragezeichenfolge.

```
Signature = URL-Encode( Base64( HMAC-SHA1( YourSecretAccessKey, UTF-8-Encoding-Of( StringToSign ) ) ) );
```

```
StringToSign = HTTP-VERB + "\n" +
  Content-MD5 + "\n" +
  Content-Type + "\n" +
  Expires + "\n" +
  CanonicalizedAmzHeaders +
  CanonicalizedResource;
```

YourSecretAccessKey ist die AWS geheime Zugriffsschlüssel-ID, die Amazon Ihnen zuweist, wenn Sie sich als Amazon Web Service-Entwickler registrieren. Beachten Sie, dass der URL-kodierte Signature ist, damit er sich für die Platzierung in der Abfragezeichenfolge eignet. Beachten Sie auch, dass in StringToSign, das HTTP Date Positionselement durch ersetzt wurde. Expires CanonicalizedAmzHeaders und CanonicalizedResource sind gleich.

Note

In der Methode für die Abfrage-String-Authentifizierung verwenden Sie den Header Date oder x-amz-date request nicht, wenn Sie die zu signierende Zeichenfolge berechnen.

Authentifizierung von Abfragezeichenfolgen-Anforderungen

Anforderung	StringToSign
<pre>GET /photos/puppy.jpg?AWSAccess KeyId=AKIAIOSFODNN7EXAMPLE& Signature=NpgCjnDzrM%2BWFzo ENXmpNDUsSn8%3D& Expires=1175139620 HTTP/1.1 Host: awsexamplebucket1.s3.us-wes t-1.amazonaws.com</pre>	<pre>GET\n \n \n 1175139620\n /awsexamplebucket1/photos/puppy.jpg</pre>

Wir gehen davon aus, dass ein Browser, wenn er die GET Anfrage stellt, weder einen Content-MD5 noch einen Content-Type-Header bereitstellt und auch keine x-amz-Header setzt, sodass diese Teile von leer bleiben. StringToSign

Verwendung der Base64-Codierung

HMACAnforderungssignaturen müssen Base64-kodiert sein. Die Base64-Kodierung konvertiert die Signatur in eine einfache ASCII Zeichenfolge, die an die Anfrage angehängt werden kann. Zeichen, die in der Signaturzeichenfolge vorkommen könnten, wie Plus (+), Schrägstrich (/) und Gleichheitszeichen (=), müssen codiert werden, wenn sie in a verwendet werden. URI Enthält beispielsweise der Authentifizierungscode ein Plussymbol (+), codieren Sie es in der Anforderung als &2B. Einen Schrägstrich codieren Sie als &2F, ein Gleichheitszeichen als %3D.

Beispiele für die Base64-Codierung finden Sie in Amazon S3 [Authentifizierungsbeispiele](#).

Browserbasierte Uploads mit POST (AWS Signaturversion 2)

Amazon S3 unterstützt POST, sodass Ihre Benutzer Inhalte direkt auf Amazon S3 hochladen können. POST wurde entwickelt, um Uploads zu vereinfachen, die Upload-Latenz zu reduzieren und Ihnen Geld für Anwendungen zu sparen, bei denen Benutzer Daten hochladen, um sie in Amazon S3 zu speichern.

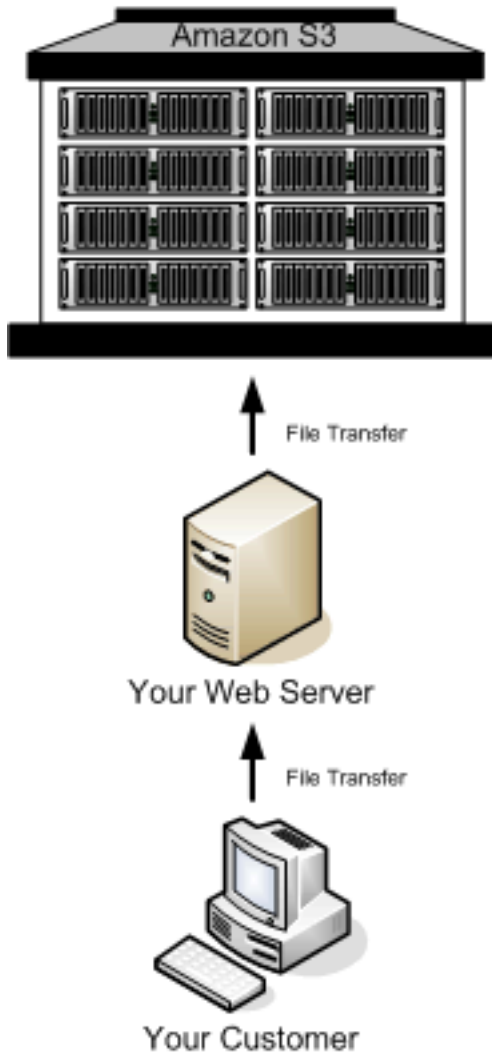
Note

Die in diesem Abschnitt beschriebene Anforderungsauthentifizierung basiert auf AWS Signature Version 2, einem Protokoll zur Authentifizierung eingehender API Anfragen an Dienste. AWS

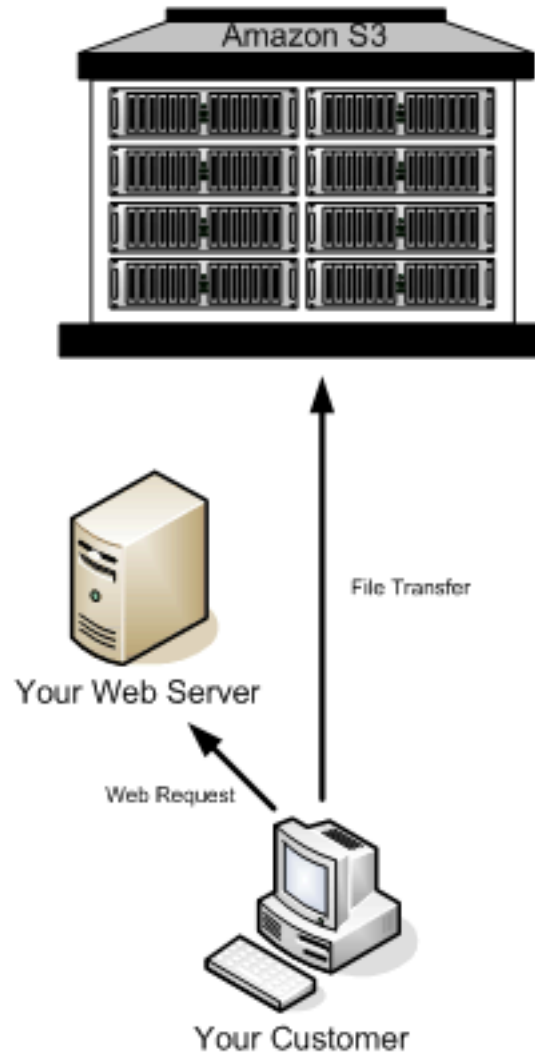
Amazon S3 unterstützt jetzt insgesamt Signature Version 4, ein Protokoll zur Authentifizierung eingehender API Anfragen an AWS Services. AWS-Regionen Derzeit wird das vor dem 30. Januar 2014 AWS-Regionen erstellte Protokoll weiterhin das vorherige Protokoll, Signature Version 2, unterstützen. Alle nach dem 30. Januar 2014 neu erstellten Regionen unterstützen nur Signature Version 4, weshalb alle Anforderungen an diese Regionen Signature Version 4 verwenden müssen. Weitere Informationen finden Sie unter [Authentifizieren von Anfragen in browserbasierten Uploads mithilfe von POST \(AWS Signature Version 4\)](#) in der Amazon Simple Storage Service-Referenz. API

Die folgende Abbildung zeigt einen Upload mit Amazon S3POST.

Proxying Amazon S3 PUTs



Using Amazon S3 POST



Hochladen mit POST

- 1 Der Benutzer öffnet einen Webbrowser, und greift auf Ihre Webseite zu.
- 2 Ihre Webseite enthält ein HTTP Formular, das alle Informationen enthält, die der Benutzer benötigt, um Inhalte auf Amazon S3 hochzuladen.
- 3 Der Benutzer lädt die Inhalte direkt zu Amazon S3 hoch.

i Note

Die Authentifizierung mit Abfragezeichenfolgen wird für nicht unterstütztPOST.

HTMLFormulare (AWS Signaturversion 2)

Themen

- [HTMLKodierung von Formularen](#)
- [HTMLFormulardeklaration](#)
- [HTMLFormularfelder](#)
- [Richtlinienerstellung](#)
- [Erstellen einer Signatur](#)
- [Umleitung](#)

Wenn Sie mit Amazon S3 kommunizieren, verwenden Sie normalerweise das REST oder, SOAP API um Put-, Get-, Delete- und andere Operationen auszuführen. Mit POST laden Benutzer Daten direkt über ihren Browser auf Amazon S3 hoch, der die Anfrage nicht verarbeiten SOAP API oder eine REST PUT Anfrage erstellen kann.

Note

SOAPSupport Over HTTP ist veraltet, aber SOAP immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Anstatt zu verwendenSOAP, empfehlen wir, entweder den REST API oder den zu verwenden AWS SDKs.

Um Benutzern das Hochladen von Inhalten auf Amazon S3 mithilfe ihrer Browser zu ermöglichen, verwenden Sie HTML Formulare. HTMLFormulare bestehen aus einer Formulardeklaration und Formularfeldern. Die Formulardeklaration enthält allgemeine Informationen zu der Anfrage. Die Formularfelder enthalten detaillierte Informationen zu der Anfragen sowie die Richtlinie, die verwendet wird, um die Anfrage zu authentifizieren und um sicherzustellen, dass sie den von Ihnen angegebenen Bedingungen entspricht.

Note

Daten und Grenzen des Formulars (ausschließlich der Inhalte der Datei) dürfen 20 KB nicht überschreiten.

In diesem Abschnitt wird die Verwendung von HTML Formularen erklärt.

HTMLKodierung von Formularen

Das Formular und die Richtlinie müssen UTF -8-kodiert sein. Sie können die Kodierung UTF -8 auf das Formular anwenden, indem Sie es in der HTML Überschrift oder als Anforderungsheader angeben.

Note

Die HTML Formulardeklaration akzeptiert keine Authentifizierungsparameter für Abfragezeichenfolgen.

Im Folgenden finden Sie ein Beispiel für die UTF -8-Kodierung in der HTML Überschrift:

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
```

Das Folgende ist ein Beispiel für die UTF -8-Kodierung in einem Anforderungsheader:

```
Content-Type: text/html; charset=UTF-8
```

HTMLFormulardeklaration

Die Formulardeklaration besteht aus drei Komponenten: der Aktion, der Methode und dem Umschließungstyp. Wenn einer dieser Werte nicht korrekt eingestellt ist, schlägt die Anforderung fehl.

Die Aktion gibt die anURL, die die Anfrage verarbeitet. Diese muss auf den Wert URL des Buckets gesetzt werden. Wenn der Name Ihres Buckets beispielsweise lautet `awsexamplebucket1` und die Region „USA West“ (Nordkalifornien) lautet, URL ist `derhttps://awsexamplebucket1.s3.us-west-1.amazonaws.com/`.

Note

Der Schlüsselname wird in einem Formularfeld angegeben.

Die Methode muss sein POST.

Der Umschließungstyp (enctype) muss angegeben werden und auf Multipart/Form-Daten für Datei- und Textbereich-Uploads gesetzt sein. Weitere Informationen finden Sie unter [RFC1867](#).

Example

Das folgende Beispiel zeigt eine Formulardeklaration für den Bucket „awsexamplebucket1“.

```
<form action="https://awsexamplebucket1.s3.us-west-1.amazonaws.com/" method="post"
enctype="multipart/form-data">
```

HTMLFormularfelder

In der folgenden Tabelle werden Felder beschrieben, die in einem HTML Formular verwendet werden können.


Note


Die Variable `${filename}` wird automatisch durch den Namen der Datei ersetzt, den der Benutzer bereitstellt, und der von allen Formularfeldern anerkannt wird. Wenn der Browser oder Client einen vollständigen oder teilweisen Pfad zu der Datei bereitstellt, wird nur der Text nach dem letzten Schrägstrich (/) oder umgekehrten Schrägstrich (\) verwendet. Beispielsweise wird „C:\Program Files\directory1\file.txt“ als „file.txt“ interpretiert. Wenn keine Datei bzw. kein Dateiname angegeben ist, wird die Variable durch eine leere Zeichenfolge ersetzt.

Feldname	Beschreibung	Erforderlich
AWSAccessKeyId	Die AWS Zugriffsschlüssel-ID des Besitzers des Buckets, der einem anonymen Benutzer	Bedingt

Feldname	Beschreibung	Erforderlich
	Zugriff auf eine Anfrage gewährt, die die Einschränkungen in der Richtlinie erfüllt. Dies ist ein Pflichtfeld, wenn die Anforderung ein Richtlinienokument beinhaltet.	
acl	<p>Eine Amazon S3 S3-Zugriffskontrollliste (ACL). Wenn eine ungültige Zugriffskontrollliste angegeben ist, wird ein Fehler generiert. Weitere Informationen zu ACLs finden Sie unter Zugriffskontrolllisten (ACLs).</p> <p>Typ: Zeichenfolge</p> <p>Standard: privat</p> <p>Zulässige Werte: private public-read public-read-write aws-exec-read authenticated-read bucket-owner-read bucket-owner-full-control</p>	Nein
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	REST-spezifische Header. Weitere Informationen finden Sie unter PUT Objekt.	Nein

Feldname	Beschreibung	Erforderlich
key	<p>Der Name des hochgeladenen Schlüssels.</p> <p>verwenden Sie die Variable <code>\$(filename)</code>, um den von dem Benutzer angegebenen Dateinamen zu verwenden. Zum Beispiel: Wenn Benutzerin „Betty“ die Datei „lolcatz.jpg“ hochlädt, und Sie <code>„/user/betty/\${filename}“</code> angeben, wird die Datei als <code>„/user/betty/lolcatz.jpg“</code> gespeichert.</p> <p>Weitere Informationen finden Sie unter Arbeiten mit Objekt-Metadaten.</p>	Ja
policy	<p>Die Sicherheitsrichtlinie, die angibt, was in der Anforderung zulässig ist. Anforderungen ohne Sicherheitsrichtlinie gelten als anonym und sind nur auf öffentlich zugänglichen Buckets erfolgreich.</p>	Nein

Feldname	Beschreibung	Erforderlich
success_action_redirect, redirect	<p>Der ClientURL, zu dem der Client bei erfolgreichem Upload umgeleitet wird. Amazon S3 hängt die Bucket-, Schlüssel- und Etag-Werte als Abfragezeichenfolgenparameter an die URL an.</p> <p>Wenn success_action_redirect nicht angegeben ist, gibt Amazon S3 den leeren Dokumententyp zurück, der im Feld success_action_status angegeben ist.</p> <p>Wenn Amazon S3 das nicht interpretieren kannURL, ignoriert es das Feld.</p> <p>Wenn der Upload fehlschlägt, zeigt Amazon S3 einen Fehler an und leitet den Benutzer nicht zu a weiterURL.</p> <p>Weitere Informationen finden Sie unter Umleitung.</p> <div data-bbox="605 1178 1268 1444" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px;"><p> Note</p><p>Der Name des Umleitungsfeldes ist veraltet, und das Feld wird künftig nicht mehr unterstützt.</p></div>	Nein

Feldname	Beschreibung	Erforderlich
success_action_status	<p>Der an den Client bei einem erfolgreichen Upload ausgegebene Statuscode, wenn success_action_redirect nicht angegeben ist.</p> <p>Gültige Werte sind 200, 201 und 204 (Standard).</p> <p>Wenn der Wert auf 200 oder 204 gesetzt ist, gibt Amazon S3 ein leeres Dokument mit dem Statuscode 200 oder 204 aus.</p> <p>Wenn der Wert auf 201 festgelegt ist, gibt Amazon S3 ein XML Dokument mit einem 201-Statuscode zurück. Informationen zum Inhalt des XML Dokuments finden Sie unter POSTObjekt.</p> <p>Wenn der Wert nicht oder auf einen falschen Wert gesetzt ist, gibt Amazon S3 ein leeres Dokument mit dem Statuscode 204 aus.</p> <div data-bbox="607 1178 1268 1631" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Einige Versionen des Adobe Flash Players verarbeiten HTTP Antworten mit leerem Text nicht richtig. Zur Unterstützung von Uploads über Adobe Flash empfehlen wir, success_action_status auf 201 zu setzen.</p></div>	Nein

Feldname	Beschreibung	Erforderlich
signature	<p>Die HMAC Signatur, die unter Verwendung des geheimen Zugriffsschlüssels erstellt wurde, der dem angegebenen entspricht AWSAccessKeyId. Dies ist ein Pflichtfeld, wenn die Anforderung ein Richtliniendokument beinhaltet.</p> <p>Weitere Informationen finden Sie unter Identity and Access Management für Amazon S3.</p>	Bedingt
x-amz-security-token	<p>Ein Sicherheitstoken, das von Sitzungsanmeldeinformationen verwendet wird</p> <p>Wenn die Anfrage Amazon DevPay verwendet, sind zwei x-amz-security-token Formularfelder erforderlich: eines für das Produkt-Token und eines für das Benutzer-Token.</p> <p>Wenn die Anforderung Sitzungsanmeldedaten verwendet, ist ein x-amz-security-token-Formular erforderlich. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldedaten im IAMBenutzerhandbuch.</p>	Nein
Andere Feldnamen mit dem x-amz-meta Präfix -	<p>Vom Benutzer angegebene Metadaten</p> <p>Amazon S3 validiert oder verwendet diese Daten nicht.</p> <p>Weitere Informationen finden Sie unter PUTObjekt.</p>	Nein

Feldname	Beschreibung	Erforderlich
file	<p>Datei oder Textinhalt.</p> <p>Die Datei oder der Inhalt muss das letzte Feld des Formulars sein. Alle Felder darunter werden ignoriert.</p> <p>Sie können nicht mehr als eine Datei zur gleichen Zeit hochladen.</p>	Ja

Richtlinienerstellung

Themen

- [Ablauf](#)
- [Bedingungen](#)
- [Übereinstimmung von Bedingungen](#)
- [Escape-Zeichen](#)

Bei der Richtlinie handelt es sich um ein UTF -8- und Base64-kodiertes JSON Dokument, das die Bedingungen festlegt, die die Anforderung erfüllen muss, und das zur Authentifizierung des Inhalts verwendet wird. Je nach dem, wie Sie Ihre Richtliniendokumente gestalten, können Sie sie pro Upload, pro Benutzer, für alle Uploads oder nach anderen Kriterien, die Ihren Anforderungen entsprechen, verwenden.

Note

Obwohl das Richtliniendokument optional ist, empfehlen nachdrücklich, es zu verwenden, anstatt einen Bucket öffentlich beschreibbar zu machen.

Nachstehend finden Sie ein Beispiel für ein Richtliniendokument:

```
{ "expiration": "2007-12-01T12:00:00.000Z",  
  
  "conditions": [
```

```
{ "acl": "public-read" },  
  
  { "bucket": "awsexamplebucket1" },  
  
  [ "starts-with", "$key", "user/eric/" ],  
  
]  
  
}
```

Das Richtliniendokument enthält den Ablauf und die Bedingungen.

Ablauf

Das Ablauelement gibt das Ablaufdatum der Richtlinie im 8601-Datumsformat an. ISO UTC
Beispielsweise gibt „2007-12-01T 12:00:00.000 Z“ an, dass die Richtlinie am 01.12.2007 nach
Mitternacht nicht gültig ist. UTC Die Angabe des Ablaufs ist in einer Richtlinie erforderlich.

Bedingungen

Die Bedingungen in dem Richtliniendokument validieren den Inhalt des hochgeladenen Objekts.
Jedes Formularfeld, das Sie im Formular angeben (mit Ausnahme von Signatur-, Datei-
AWSAccessKeyId, Richtlinien- und Feldnamen, die ein x-ignore- Präfix haben), muss in der Liste der
Bedingungen enthalten sein.

Note

Wenn Sie mehrere Felder mit dem gleichen Namen haben, müssen die Werte durch
Kommata abgeteilt sein. Wenn Sie beispielsweise zwei Felder mit dem Namen "x-amz-meta-
tag" haben und das erste den Wert „Ninja“ und das zweite den Wert „Stallman“ hat, würden
Sie das Richtliniendokument auf einstellen. Ninja, Stallman

Alle Variablen in dem Formular werden vor der Validierung der Richtlinie erweitert. Daher
müssen alle Bedingungsabgleiche anhand der erweiterten Felder vorgenommen werden.
Zum Beispiel: Wenn Sie das Schlüsselfeld auf user/betty/\${filename} setzen, ist Ihre
Richtlinie möglicherweise ["starts-with", "\$key", "user/betty/"]. Geben
Sie nicht ei ["starts-with", "\$key", "user/betty/\${filename}"]. Weitere
Informationen finden Sie unter [Übereinstimmung von Bedingungen](#).

Die folgende Tabelle beschreibt die Bedingungen für Richtliniendokumente.

Elementname	Beschreibung
acl	Gibt die Bedingungen an, die erfüllt werden müssenACL. Unterstützt exakte Übereinstimmung und <code>starts-with</code> .
content-length-range	Gibt die erlaubte Mindest- und Höchstgröße des hochgeladenen Inhalts an. Unterstützt den Bereichsabgleich.
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	REST-spezifische Header. Unterstützt exakte Übereinstimmung und <code>starts-with</code> .
Schlüssel	Der Name des hochgeladenen Schlüssels. Unterstützt exakte Übereinstimmung und <code>starts-with</code> .
success_action_redirect, redirect	DerURL, zu dem der Client bei erfolgreichem Upload umgeleitet wird. Unterstützt exakte Übereinstimmung und <code>starts-with</code> .
success_action_status	Der an den Client bei einem erfolgreichen Upload ausgegebene Statuscode, wenn <code>success_action_redirect</code> nicht angegeben ist. Unterstützt exakte Übereinstimmung.
x-amz-security-token	DevPay Amazon-Sicherheitstoken. Für jede Anfrage, die Amazon verwendet, DevPay sind zwei <code>x-amz-security-token</code> Formularfelder erforderlich:

Elementname	Beschreibung
	eines für das Produkt-Token und eines für das Benutzer-Token. Daher müssen die Werte durch Kommata voneinander getrennt werden. Zum Beispiel: Wenn der Benutzer-Token <code>eW91dHViZQ==</code> und der Produkt-Token <code>b0hnNVNKWVJIQTA=</code> ist, setzen Sie den Richtlinieneintrag auf: <pre>{ "x-amz-security-token": "eW91dHViZQ==,b0hnNVNKWVJIQTA=" }</pre>
Andere Feldnamen mit dem <code>x-amz-meta</code> Präfix -	Vom Benutzer angegebene Metadaten Unterstützt exakte Übereinstimmung und <code>starts-with</code> .

Note

Wenn Ihr Toolkit weitere Felder hinzufügt (beispielsweise fügt Flash den Dateinamen hinzu), müssen Sie diese dem Richtliniendokument hinzufügen. Wenn Sie diese Funktionalität steuern können, setzen Sie `x-ignore-` vor das Feld, sodass Amazon S3 die Funktion ignoriert und künftige Versionen davon unbeeinflusst lässt.

Übereinstimmung von Bedingungen

Die folgende Tabelle beschreibt die Übereinstimmungstypen von Bedingungen. Obwohl Sie für jedes Formularfeld, das Sie in dem Formular angeben, eine Bedingung angeben müssen, können Sie auch komplexere Übereinstimmungskriterien erstellen, indem Sie für ein Formularfeld mehrere Bedingungen angeben.

Bedingung	Beschreibung
Genauere Übereinstimmung	Hierbei müssen die Felder spezifische Werte enthalten. Dieses Beispiel zeigt, dass der auf <code>public-read</code> gesetzt werden ACL muss: <pre>{"acl": "public-read" }</pre>

Bedingung	Beschreibung
	<p>Dieses Beispiel ist eine alternative Methode, um anzugeben, dass für auf public-read gesetzt werden ACL muss:</p> <pre data-bbox="375 331 1507 411">["eq", "\$acl", "public-read"]</pre>
Beginnt mit	<p>Verwenden Sie diese Bedingung, wenn der Wert mit einem bestimmten Wert beginnen muss. In diesem Beispiel wird angegeben, dass der Schlüssel mit „user/betty“ beginnen muss:</p> <pre data-bbox="375 678 1507 758">["starts-with", "\$key", "user/betty/"]</pre>
Übereinstimmung mit beliebigem Inhalt	<p>Verwenden Sie „starts-with“ mit einem leeren Wert, um die Richtlinie so zu konfigurieren, dass in einem Feld beliebiger Inhalt zulässig ist. Dieses Beispiel lässt jeden Wert für success_action_redirect zu:</p> <pre data-bbox="375 1020 1507 1100">["starts-with", "\$success_action_redirect", ""]</pre>
Angabe von Bereichen	<p>Wenn Felder Bereiche akzeptieren, trennen Sie die Ober- und die Untergrenze durch ein Komma voneinander ab. In diesem Beispiel sind Dateigrößen von 1 bis 10 Megabyte erlaubt:</p> <pre data-bbox="375 1367 1507 1446">["content-length-range", 1048579, 10485760]</pre>

Escape-Zeichen

Die folgende Tabelle beschreibt Zeichen, für die in einem Richtliniendokument ein Escape-Zeichen verwendet werden muss.

Escape-Sequenz	Beschreibung
\\	Umgekehrter Schrägstrich
\\\$	Dollarzeichen
\\b	Backspace
\\f	Seitenvorschub
\\n	Neue Zeile
\\r	Zeilenumschaltung
\\t	Horizontaler Tabulator
\\V	Vertikaler Tabulator
\\u <code>xxxx</code>	Alle Unicode-Zeichen

Erstellen einer Signatur

Schritt	Beschreibung
1	Kodieren Sie die Richtlinie mit UTF -8.
2	Kodieren Sie diese UTF -8 Byte mit Base64.
3	

Schritt	Beschreibung
	Signieren Sie die Richtlinie mit Ihrem geheimen Zugriffsschlüssel, indem Sie -1 verwenden. HMAC SHA
4	Kodieren Sie die Signatur SHA -1 mit Base64.

Allgemeine Informationen über die Authentifizierung finden Sie unter [Identity and Access Management für Amazon S3](#).

Umleitung

In diesem Abschnitt wird beschrieben, wie Sie mit Umleitungen umgehen.

Allgemeine Umleitung

Nach Abschluss der POST Anfrage wird der Benutzer an den Ort weitergeleitet, den Sie in dem `success_action_redirect` Feld angegeben haben. Wenn Amazon S3 das nicht interpretieren kann URL, ignoriert es das `success_action_redirect` Feld.

Wenn `success_action_redirect` nicht angegeben ist, gibt Amazon S3 den leeren Dokumententyp zurück, der im Feld `success_action_status` angegeben ist.

Wenn die POST Anfrage fehlschlägt, zeigt Amazon S3 einen Fehler an und stellt keine Weiterleitung bereit.

Umleitung vor dem Upload

Wenn Ihr Bucket mit `< CreateBucketConfiguration >` erstellt wurde, benötigen Ihre Endbenutzer möglicherweise eine Umleitung. Wenn dies der Fall ist, kann es sein, dass einige Browser mit der Umleitung nicht korrekt umgehen. Dies kommt relativ selten vor, meistens dann, wenn der Bucket gerade eben erstellt wurde.

Laden Sie Beispiele hoch (AWS Signaturversion 2)

Themen

- [Datei-Upload](#)
- [Textbereich-Upload](#)

Note

Die in diesem Abschnitt beschriebene Anforderungsauthentifizierung basiert auf AWS Signature Version 2, einem Protokoll zur Authentifizierung eingehender API Anfragen an AWS Dienste.

Amazon S3 unterstützt jetzt insgesamt Signature Version 4, ein Protokoll zur Authentifizierung eingehender API Anfragen an AWS Services. AWS-Regionen Derzeit wird das vor dem 30. Januar 2014 AWS-Regionen erstellte Protokoll weiterhin das vorherige Protokoll, Signature Version 2, unterstützen. Alle nach dem 30. Januar 2014 neu erstellten Regionen unterstützen nur Signature Version 4, weshalb alle Anforderungen an diese Regionen Signature Version 4 verwenden müssen. Weitere Informationen finden Sie unter [Beispiele: Browserbasierter Upload mit HTTP POST \(unter Verwendung von AWS Signature Version 4\)](#) in der Amazon Simple Storage API Service-Referenz.

Datei-Upload

Dieses Beispiel zeigt den vollständigen Vorgang der Erstellung einer Richtlinie und eines Formulars für den Upload eines Dateianhangs.

Erstellung von Richtlinie und Formular

Die folgende Richtlinie unterstützt Uploads zu Amazon S3 für den Bucket „awsexamplebucket1“.

```
{ "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "awsexamplebucket1"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
    {"success_action_redirect": "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html"},
    ["starts-with", "$Content-Type", "image/"],
    {"x-amz-meta-uuid": "14365123651274"},
    ["starts-with", "$x-amz-meta-tag", ""]
  ]
}
```

Für diese Richtlinie ist Folgendes erforderlich:

- Der Upload muss vor 12:00 Uhr UTC am 1. Dezember 2007 erfolgen.

- Der Inhalt muss zum Bucket „awsexamplebucket1“ hochgeladen werden.
- Der Schlüssel muss mit „user/eric/“ beginnen.
- Das ACL ist auf öffentlich lesbar eingestellt.
- Der Success_Action_Redirect ist auf gesetzt. https://awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html
- Bei dem Objekt handelt es sich um eine Bilddatei.
- Das x-amz-meta-uuid Tag muss auf 14365123651274 gesetzt sein.
- Das x-amz-meta-tag kann einen beliebigen Wert enthalten.

Nachfolgend sehen Sie eine mit Base64 kodierte Version dieser Richtlinie.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTExVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwoGICAgcyJidW
```

Erstellen Sie unter Verwendung Ihrer Anmeldedaten eine Signatur, zum Beispiel ist `0RavWzkygo6QX9caELEqKi9kDbU=` die Signatur für das vorangehende Richtliniendokument.

Das folgende Formular unterstützt eine POST Anfrage an den `amzn-s3-demo-bucket` Bucket, der diese Richtlinie verwendet.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
    ...
    <form action="https://amzn-s3-demo-bucket.s3.us-west-1.amazonaws.com/" method="post"
  enctype="multipart/form-data">
      Key to upload: <input type="input" name="key" value="user/eric/" /><br />
      <input type="hidden" name="acl" value="public-read" />
      <input type="hidden" name="success_action_redirect" value="https://
  awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html" />
      Content-Type: <input type="input" name="Content-Type" value="image/jpeg" /><br />
      <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />
      Tags for File: <input type="input" name="x-amz-meta-tag" value="" /><br />
      <input type="hidden" name="AWSAccessKeyId" value="AKIAIOSFODNN7EXAMPLE" />
```

```
<input type="hidden" name="Policy" value="POLICY" />
<input type="hidden" name="Signature" value="SIGNATURE" />
File: <input type="file" name="file" /> <br />
<!-- The elements after this will be ignored -->
<input type="submit" name="submit" value="Upload to Amazon S3" />
</form>
...
</html>
```

Beispielanforderung

Diese Anforderung geht davon aus, dass das hochgeladene Bild 117.108 Bytes groß ist, die Bilddaten nicht eingeschlossen.

```
POST / HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.10) Gecko/20071115
  Firefox/2.0.0.10
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=9431149156168
Content-Length: 118698

--9431149156168
Content-Disposition: form-data; name="key"

user/eric/MyPicture.jpg
--9431149156168
Content-Disposition: form-data; name="acl"

public-read
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"

https://awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html
--9431149156168
Content-Disposition: form-data; name="Content-Type"
```

```

image/jpeg
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-uuid"

14365123651274
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"

Some, Tag, For, Picture
--9431149156168
Content-Disposition: form-data; name="AWSAccessKeyId"

AKIAIOSFODNN7EXAMPLE
--9431149156168
Content-Disposition: form-data; name="Policy"

eyJhZXBhYXN0eXhwaXJhdGlvbiI6IClYMDA3LTExVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgcyJidWN
--9431149156168
Content-Disposition: form-data; name="Signature"

0RavWzkygo6QX9caELEqKi9kDbU=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg

...file content...
--9431149156168
Content-Disposition: form-data; name="submit"

Upload to Amazon S3
--9431149156168--

```

Beispielantwort

```

HTTP/1.1 303 Redirect
x-amz-request-id: 1AEE782442F35865
x-amz-id-2: cxzFLJRatFHy+NGtaDFRR8YvI9BHmgLxjvJzNiGGICARZ/mVXHj7T+qQKhdpzHFh
Content-Type: application/xml
Date: Wed, 14 Nov 2007 21:21:33 GMT
Connection: close
Location: https://awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html?bucket=awsexamplebucket1&key=user/eric/MyPicture.jpg&etag="39d459dfbc0faabbb5e179358dfb94c3";

```

Server: AmazonS3

Textbereich-Upload

Themen

- [Erstellung von Richtlinie und Formular](#)
- [Beispielanforderung](#)
- [Beispielantwort](#)

Das folgende Beispiel zeigt den vollständigen Prozess der Erstellung einer Richtlinie und eines Formulars für den Upload eines Textbereiches. Der Upload eines Textbereiches ist nützlich für die Übermittlung von Benutzern erstellter Inhalte, etwa von Blog-Postings.

Erstellung von Richtlinie und Formular

Die folgende Richtlinie unterstützt das Hochladen von Textbereichen zu Amazon S3 für den Bucket „awsexamplebucket1“.

```
{ "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "awsexamplebucket1"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
    {"success_action_redirect": "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html"},
    ["eq", "$Content-Type", "text/html"],
    {"x-amz-meta-uuid": "14365123651274"},
    ["starts-with", "$x-amz-meta-tag", ""]
  ]
}
```

Für diese Richtlinie ist Folgendes erforderlich:

- Der Upload muss GMT am 01.12.2007 vor 12:00 Uhr erfolgen.
- Der Inhalt muss zum Bucket „awsexamplebucket1“ hochgeladen werden.
- Der Schlüssel muss mit „user/eric/“ beginnen.
- Das ACL ist auf öffentlich lesbar eingestellt.
- Der Success_Action_Redirect ist auf gesetzt. https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html

Your blog post goes here.

```
</textarea><br />
<!-- The elements after this will be ignored -->
<input type="submit" name="submit" value="Upload to Amazon S3" />
</form>
...
</html>
```

Beispielanforderung

Diese Anforderung geht davon aus, dass das hochgeladene Bild 117.108 Bytes groß ist, die Bilddaten nicht eingeschlossen.

```
POST / HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.10) Gecko/20071115
  Firefox/2.0.0.10
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=178521717625888
Content-Length: 118635

-178521717625888
Content-Disposition: form-data; name="key"

ser/eric/NewEntry.html
--178521717625888
Content-Disposition: form-data; name="acl"

public-read
--178521717625888
Content-Disposition: form-data; name="success_action_redirect"

https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html
--178521717625888
Content-Disposition: form-data; name="Content-Type"
```


POST mit Adobe Flash

In diesem Abschnitt wird die Verwendung von POST mit Adobe Flash beschrieben.

Adobe Flash Player-Sicherheit

Standardmäßig verbietet das Sicherheitsmodell von Adobe Flash Player, Netzwerkverbindungen zu Servern außerhalb der Domäne herzustellen, die die Datei bereitstellt. SWF

Um die Standardeinstellung zu überschreiben, müssen Sie eine öffentlich lesbare Datei `crossdomain.xml` in den Bucket hochladen, der POST Uploads akzeptiert. Nachfolgend sehen Sie ein Beispiel für eine solche `crossdomain.xml`-Datei.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<allow-access-from domain="*" secure="false" />
</cross-domain-policy>
```

Note

Weitere Informationen zum Sicherheitsmodell von Adobe Flash finden Sie auf der Adobe-Website.

Durch die Hinzufügung der `crossdomain.xml`-Datei kann jeder Adobe Flash Player eine Verbindung zu der `crossdomain.xml`-Datei in Ihrem Bucket herstellen; diese Datei gewährt jedoch keinen Zugriff auf den Amazon-S3-Bucket selbst.

Überlegungen zu Adobe Flash

FileReference API In Adobe Flash wird das `Filename` Formularfeld zur POST Anfrage hinzugefügt. Wenn Sie Adobe Flash-Anwendungen erstellen, die mithilfe der FileReference API Aktion auf Amazon S3 hochgeladen werden, nehmen Sie die folgende Bedingung in Ihre Richtlinie auf:

```
['starts-with', '$Filename', '']
```

Einige Versionen von Adobe Flash Player verarbeiten HTTP Antworten mit leerem Text nicht richtig. Um so POST zu konfigurieren, dass eine Antwort zurückgegeben wird, die keinen leeren Text enthält, legen Sie `success_action_status` den Wert 201 fest. Amazon S3 sendet dann ein XML

Dokument mit einem 201-Statuscode zurück. Informationen zum Inhalt des XML Dokuments finden Sie unter [POSTObjekt](#). Informationen zu den Formularfeldern finden Sie unter [HTMLFormularfelder](#).

Bewährte Methoden für Designmuster: Optimieren der Leistung von Amazon S3

Ihre Anwendungen erreichen schnell Tausende von Transaktionen pro Sekunde bei der Anfrageleistung, wenn Speicherinhalte zu Amazon S3 hochgeladen oder von dort abgerufen werden. Amazon S3 wird automatisch auf hohe Anfrageraten skaliert. Ihre Anwendung kann beispielsweise mindestens 3.500 PUT/DELETE oder 5.500 COPY POSTGET/HEAD Anfragen pro Sekunde pro partitioniertem Amazon S3 S3-Präfix erreichen. Es gibt keine Einschränkungen für die Anzahl der Präfixe in einem Bucket. Sie können Ihre Lese- und Schreibleistung steigern, indem Sie Parallelisierung durchführen. Wenn Sie beispielsweise 10 Präfixe in einem Amazon S3-Bucket für parallele Lesevorgänge einrichten, können Sie damit die Leseleistung auf 55 000 Leseanfragen pro Sekunde skalieren. Auf ähnliche Weise können Sie Schreibvorgänge skalieren, indem Sie auf mehrere Präfixe schreiben. Die Skalierung erfolgt sowohl bei Lese- als auch bei Schreiboperationen schrittweise und nicht sofort. Während Amazon S3 auf Ihre neue höhere Anforderungsrate skaliert, können einige 503 (Slow Down)-Fehler auftreten. Diese Fehler werden nicht mehr angezeigt, sobald die Skalierung abgeschlossen ist. Weitere Informationen zum Erstellen und Verwenden von Präfixen finden Sie unter [Organisieren von Objekten mit Präfixen](#).

So prüfen etwa manche Data Lake-Anwendungen in Amazon S3 Millionen oder Milliarden von Objekten auf Anfragen, die über Petabytes von Daten ausgeführt werden. Diese Data-Lake-Anwendungen erzielen Single-Instance-Übertragungsraten, die die Netzwerkschnittstellennutzung für ihre [EC2 Amazon-Instance](#) maximieren. Diese kann bis zu 100 GB/s auf einer einzelnen Instance betragen. Anschließend aggregieren diese Anwendungen den Durchsatz über mehrere Instances hinweg, um mehrere Terabit pro Sekunde zu erreichen.

Andere Anwendungen sind latenzempfindlich, wie etwa Social-Media-Messaginganwendungen. Diese Anwendungen können konsistente Latenzen für kleine Objekte (und first-byte-out Latenzen für größere Objekte) von etwa 100 bis 200 Millisekunden erreichen.

Sonstige AWS Services können auch dazu beitragen, die Leistung verschiedener Anwendungsarchitekturen zu beschleunigen. Wenn Sie beispielsweise höhere Übertragungsraten über eine einzelne HTTP Verbindung oder Latenzen im einstelligen Millisekundenbereich wünschen, verwenden Sie [Amazon CloudFront oder Amazon ElastiCache](#) für das Caching mit [Amazon S3](#).

Und wenn Sie eine schnelle Datenübertragung über große Distanzen zwischen einem Client und einem S3-Bucket wünschen, verwenden Sie [Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#). Transfer Acceleration nutzt die global verteilten Edge-Standorte,

um den Datentransport über geografische CloudFront Entfernungen zu beschleunigen. Wenn Ihr Amazon S3 S3-Workload serverseitige Verschlüsselung verwendet mit AWS KMS, siehe [AWS KMS Grenzen](#) in der AWS Key Management Service Im Entwicklerhandbuch finden Sie Informationen zu den für Ihren Anwendungsfall unterstützten Anforderungsraten.

Die folgenden Themen beschreiben bewährte Verfahren und Designmuster zur Optimierung der Leistung von Anwendungen, die Amazon S3 verwenden. Die neuesten Informationen zur Leistungsoptimierung für Amazon S3 finden Sie unter [Anleitungen zur Leistung von Amazon S3](#) und [Leistungsdesignmuster für Amazon S3](#).

Note

Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter [Was ist S3 Express One Zone?](#) und [Verzeichnis-Buckets](#).

Themen

- [Anleitungen zur Leistung von Amazon S3](#)
- [Leistungsdesignmuster für Amazon S3](#)

Anleitungen zur Leistung von Amazon S3

Bei der Erstellung von Anwendungen, die Objekte zu Amazon S3 hochladen und davon abrufen, sollten Sie unsere bewährten Methoden befolgen, um die Leistung zu optimieren. Wir bieten auch detailliertere [Leistungsdesignmuster](#).

Zur Erzielung der besten Leistung für Ihre Anwendung auf Amazon S3 empfehlen wir die folgenden Vorgehensweisen.

Themen

- [Messen der Leistung](#)
- [Horizontale Skalierung von Speicherverbindungen](#)
- [Verwenden von Byte Range Fetches](#)
- [Wiederholungsanforderungen für latenzsensitive Anwendungen](#)

- [Kombinieren Sie Amazon S3 \(Storage\) und Amazon EC2 \(Compute\) in einem AWS-Region](#)
- [Verwenden von Amazon S3 Transfer Acceleration zur Minimierung der durch die Entfernung verursachten Latenz](#)
- [Verwenden Sie die neueste Version von AWS SDKs](#)

Messen der Leistung

Achten Sie bei der Leistungsoptimierung auf den Netzwerkdurchsatz und die DRAM Anforderungen. CPU Abhängig von der Mischung der Anforderungen an diese verschiedenen Ressourcen kann es sich lohnen, verschiedene [EC2Amazon-Instance-Typen](#) zu evaluieren. Weitere Informationen zu Instance-Typen finden Sie unter [Instance-Typen](#) im EC2Amazon-Benutzerhandbuch.

Es ist auch hilfreich, bei der Leistungsmessung die DNS Suchzeit, die Latenz und die Datenübertragungsgeschwindigkeit mithilfe von HTTP Analysetools zu berücksichtigen.

Um die Leistungsanforderungen zu verstehen und die Leistung Ihrer Anwendung zu optimieren, können Sie auch die 503-Fehlerantworten überwachen, die Sie erhalten. Die Überwachung bestimmter Leistungsmetriken kann mit zusätzlichen Kosten verbunden sein. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Überwachen der Anzahl der Antworten auf 503 (Slow Down)-Statusfehler

Um die Anzahl der Antworten auf 503-Statusfehler zu überwachen, die Sie erhalten, können Sie eine der folgenden Optionen verwenden:

- Verwenden Sie CloudWatch Amazon-Anforderungsmetriken für Amazon S3. Die CloudWatch Anforderungsmetriken beinhalten eine Metrik für Antworten mit dem Status 5xx. Weitere Informationen zu CloudWatch Anforderungsmetriken finden Sie unter [Metriken mit Amazon überwachen CloudWatch](#).
- Verwenden Sie die Anzahl der 503 (Service Unavailable)-Fehler, die im Abschnitt „Erweiterte Metriken“ von Amazon S3 Storage Lens verfügbar ist. Weitere Informationen finden Sie unter [Verwenden der Metriken von S3 Storage Lens zur Verbesserung der Leistung](#).
- Verwenden Sie die Amazon-S3-Serverzugriffsprotokollierung. Mit der Serverzugriffsprotokollierung können Sie alle Anforderungen filtern und überprüfen, die 503 (Internal Error)-Antworten erhalten. Sie können auch Amazon Athena verwenden, um Protokolle zu analysieren. Weitere Informationen zu Server-Zugriffsprotokollen finden Sie unter [Protokollieren von Anfragen mit Server-Zugriffsprotokollierung](#).

Wenn Sie die Anzahl der Statusfehlercodes HTTP 503 überwachen, können Sie häufig wertvolle Erkenntnisse darüber gewinnen, welche Präfixe, Schlüssel oder Buckets die Anfragen am stärksten einschränken.

Horizontale Skalierung von Speicherverbindungen

Die Verteilung von Anfragen über zahlreiche Verbindungen ist ein übliches Designmuster für die horizontale Skalierung der Leistung. Wenn Sie hochleistungsfähige Anwendungen erstellen, stellen Sie sich Amazon S3 als sehr großes verteiltes System vor, nicht als einen einzelnen Netzwerkendpunkt wie ein herkömmlicher Speicherserver. Sie erreichen die beste Leistung durch die Ausgabe mehrerer gleichzeitiger Anfragen an Amazon S3. Verteilen Sie diese Anfragen über separate Verbindungen, um die von Amazon S3 aus zugängliche Bandbreite zu maximieren. Amazon S3 hat keine Beschränkungen für die Anzahl der Verbindungen, die mit Ihrem Bucket hergestellt werden.

Verwenden von Byte Range Fetches

Mithilfe des Range HTTP Headers in einer [GETObjektanforderung](#) können Sie einen Bytebereich von einem Objekt abrufen und dabei nur den angegebenen Teil übertragen. Sie können gleichzeitige Verbindungen zu Amazon S3 verwenden, um verschiedene Bytebereiche aus demselben Objekt abzurufen. Dies hilft beim Erreichen eines höheren aggregierten Durchsatzes als bei einer einzelnen Anforderung eines ganzen Objekts. Der Abruf kleinerer Bereiche eines größeren Objekts ermöglicht Ihrer Anwendung auch die Verbesserung der Zeiten für Wiederholungsversuche, wenn Anforderungen unterbrochen werden. Weitere Informationen finden Sie unter [Herunterladen von Objekten](#).

Typische Größen für Bytebereichsanforderungen sind 8 oder 16 MB. Wenn Objekte einen mehrteiligen Upload PUT verwenden, empfiehlt es sich, GET sie in den gleichen Teilgrößen (oder zumindest an den Teilgrenzen) zu platzieren, um eine optimale Leistung zu erzielen. GETAnfragen können sich direkt auf einzelne Teile beziehen, zum Beispiel `GET ?partNumber=N`.

Wiederholungsanforderungen für latenzsensitive Anwendungen

Aggressive Timeouts und Wiederholungsversuche sorgen für konsistente Latenz. Aufgrund des Umfangs von Amazon S3 gilt: Wenn die erste Anforderung langsam ist, nimmt eine alte Anfrage wahrscheinlich einen anderen Pfad und ist schnell erfolgreich. Das Tool AWS SDKs verfügen über konfigurierbare Werte für Timeout und Wiederholungsversuche, die Sie an die Toleranzen Ihrer spezifischen Anwendung anpassen können.

Kombinieren Sie Amazon S3 (Storage) und Amazon EC2 (Compute) in einem AWS-Region

Obwohl Namen von S3-Buckets [global eindeutig](#) sind, wird jeder Bucket in einer Region gespeichert, die Sie bei dessen Erstellung auswählen. Um die Leistung zu optimieren, empfehlen wir, dass Sie auf den Bucket von EC2 Amazon-Instances aus in derselben Weise zugreifen AWS-Region wenn möglich. Dies hilft bei der Reduzierung der Netzwerklatenz und der Datenübertragungskosten.

Weitere Informationen zu den Kosten von Datenübertragungen finden Sie unter [Amazon S3 – Preise](#).

Verwenden von Amazon S3 Transfer Acceleration zur Minimierung der durch die Entfernung verursachten Latenz

[Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#)

ermöglicht die schnelle, einfache und sichere Übertragung von Dateien über größere Entfernungen zwischen Ihrem Client und einem S3-Bucket. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte von [Amazon CloudFront](#). Sobald die Daten an einem Edge-Standort eingeht, werden sie über einen optimierten Netzwerkpfad an Amazon S3 weitergeleitet. Transfer Acceleration ist ideal für die regelmäßige Übertragung von Daten im Gigabyte- bis Terabyte-Bereich von Kontinent zu Kontinent geeignet. Die Funktion ist auch für Kunden nützlich, die Uploads in einen zentralen Bucket aus der ganzen Welt vornehmen.

Sie können das [Amazon S3 Transfer Acceleration Speed Comparison Tool](#) verwenden, um beschleunigte und nicht beschleunigte Upload-Geschwindigkeiten in allen Amazon S3-Regionen zu vergleichen. Das Speed Comparison-Tool verwendet mehrteilige Uploads, um eine Datei von Ihrem Browser in verschiedene Amazon S3-Regionen mit und ohne Amazon S3 Transfer Acceleration zu übertragen.

Verwenden Sie die neueste Version von AWS SDKs

Das Tool AWS SDKs bieten integrierten Support für viele der empfohlenen Richtlinien zur Optimierung der Amazon S3 S3-Leistung. SDKs Sie bieten eine API einfachere Nutzung der Vorteile von Amazon S3 innerhalb einer Anwendung und werden regelmäßig aktualisiert, um den neuesten Best Practices zu entsprechen. Sie SDKs enthalten beispielsweise eine Logik, mit der Anfragen bei HTTP 503-Fehlern automatisch wiederholt werden, und investieren in Code, um auf langsame Verbindungen zu reagieren und sich an diese anzupassen.

Sie bieten SDKs auch den [Transfer Manager](#), der die horizontale Skalierung von Verbindungen automatisiert, um Tausende von Anfragen pro Sekunde zu erreichen, wobei gegebenenfalls Anfragen

im Bytebereich verwendet werden. Es ist wichtig, die neueste Version von zu verwenden AWS SDKsum die neuesten Funktionen zur Leistungsoptimierung zu erhalten.

Sie können die Leistung auch optimieren, wenn Sie HTTP REST API Anfragen verwenden. Bei der REST API Verwendung von sollten Sie dieselben bewährten Methoden befolgen, die Teil von sind SDKs. Lassen Sie bei langsamen Anforderungen Timeouts und Wiederholungsversuche zu, sowie mehrere Verbindung, um den parallelen Abruf von Objektdaten zu ermöglichen. Informationen zur Verwendung von finden Sie in der [Amazon Simple Storage Service API Reference](#). REST API

Leistungsdesignmuster für Amazon S3

Beim Entwurf von Anwendungen zum Upload und Abruf von Objekten von Amazon S3 sollten Sie unsere bewährten Designmuster verwenden, um eine optimale Leistung für Ihre Anwendung zu erzielen. Dazu bieten wir Ihnen [Leistungsanleitungen](#) für Überlegungen zur Planung Ihrer Anwendungsarchitektur an.

Zur Optimierung der Leistung können Sie die folgenden Designmuster verwenden.

Themen

- [Verwendung von Caching für Inhalte mit häufigen Zugriffen](#)
- [Timeouts und Wiederholungsversuche für latenzsensitive Anwendungen](#)
- [Horizontale Skalierung und Anforderungsparallelisierung für hohen Durchsatz](#)
- [Verwendung von Amazon S3 Transfer Acceleration zur Beschleunigung geographisch disparater Datenübertragungen](#)

Verwendung von Caching für Inhalte mit häufigen Zugriffen

Viele Anwendungen, die Daten in Amazon S3 speichern, stellen einen „Arbeitssatz“ der Daten bereit, der von Benutzern häufig angefragt wird. Wenn ein Workload wiederholte GET Anfragen für eine gemeinsame Gruppe von Objekten sendet, können Sie einen Cache wie [Amazon CloudFront](#) ElastiCache, [Amazon](#) oder [AWS Elemental MediaStore](#)um die Leistung zu optimieren. Die erfolgreiche Cache-Nutzung kann zu niedriger Latenz und zu hohen Datenübertragungsraten führen. Anwendungen, die die Zwischenspeicherung verwenden, senden auch weniger direkte Anforderungen an Amazon S3, was zur Senkung der Anfragekosten beitragen kann.

Amazon CloudFront ist ein schnelles Netzwerk zur Bereitstellung von Inhalten (CDN), das Daten aus Amazon S3 transparent in einer Vielzahl von geografisch verteilten Präsenzpunkten

() zwischenspeichert. PoPs Wenn auf Objekte aus mehreren Regionen oder über das Internet zugegriffen werden kann, CloudFront können Daten in der Nähe der Benutzer zwischengespeichert werden, die auf die Objekte zugreifen. Dies kann zu hoher Leistung bei der Bereitstellung beliebiger Amazon S3-Inhalte führen. Informationen dazu CloudFront finden Sie im [Amazon CloudFront Developer Guide](#).

Amazon ElastiCache ist ein verwalteter In-Memory-Cache. Mit ElastiCache können Sie EC2 Amazon-Instances bereitstellen, die Objekte im Speicher zwischenspeichern. Dieses Caching führt zu einer Verringerung der GET Latenz um ein Vielfaches und zu einer erheblichen Steigerung des Download-Durchsatzes. Um dies zu verwenden ElastiCache, ändern Sie die Anwendungslogik so, dass sowohl der Cache mit Hot-Objekten gefüllt wird als auch der Cache auf Hot-Objekte überprüft wird, bevor Sie sie von Amazon S3 anfordern. Beispiele für die Verwendung ElastiCache zur Verbesserung der GET Leistung von Amazon S3 finden Sie im Blogbeitrag [Turbocharge Amazon S3 with Amazon ElastiCache for Redis](#).

AWS Elemental MediaStore ist ein Caching- und Inhaltsverteilungssystem, das speziell für Video-Workflows und Medienbereitstellung von Amazon S3 entwickelt wurde. MediaStore bietet end-to-end Speicher APIs speziell für Videos und wird für leistungsabhängige Video-Workloads empfohlen. Informationen MediaStore zu finden Sie im [AWS Elemental MediaStore Benutzerleitfaden](#).

Timeouts und Wiederholungsversuche für latenzsensitive Anwendungen

Es gibt bestimmte Situationen, in denen eine Anwendung eine Antwort von Amazon S3 erhält, die darauf hinweist, dass ein Wiederholungsversuch erforderlich ist. Amazon S3 ordnet Bucket- und Objektnamen den damit verbundenen Objektdaten zu. Wenn eine Anwendung hohe Anforderungsraten generiert (in der Regel anhaltende Raten von über 5.000 Anfragen pro Sekunde an eine kleine Anzahl von Objekten), erhält sie möglicherweise HTTP 503 verlangsamte Antworten. Wenn diese Fehler auftreten, wird jeweils AWS SDK implementiert eine automatische Wiederholungslogik unter Verwendung von exponentiellem Backoff. Wenn Sie kein AWS SDK, sollten Sie die Wiederholungslogik implementieren, wenn Sie den HTTP 503-Fehler erhalten. Informationen zu Backoff-Techniken finden Sie unter [Error Retries and Exponential Backoff in AWS](#) in der Allgemeine Amazon Web Services-Referenz.

Amazon S3 wird automatisch in Reaktion auf andauernde neue Anforderungsraten skaliert und optimiert so die Leistung in dynamischer Weise. Während Amazon S3 intern für eine neue Anforderungsrate optimiert, erhalten Sie vorübergehend HTTP 503 Antworten auf Anfragen, bis die Optimierung abgeschlossen ist. Nachdem Amazon S3 die Leistung intern für die neue Anfragerate optimiert hat, werden alle Anfragen generell ohne Wiederholungsversuche bereitgestellt.

Für latenzsensitive Anwendungen empfiehlt Amazon S3 Nachverfolgung und aggressive Wiederholungsversuche bei langsameren Vorgängen. Wenn Sie eine Anfrage erneut versuchen, empfehlen wir, eine neue Verbindung zu Amazon S3 zu verwenden und eine neue DNS Suche durchzuführen.

Wenn Sie sehr große Anforderungen unterschiedlicher Größe (beispielsweise mit mehr als 128 MB) durchführen, sollten Sie den erreichten Durchsatz nachverfolgen und für die langsamsten 5 Prozent der Anforderungen Wiederholungsversuche durchführen. Wenn Sie kleinere Anfragen stellen (z. B. weniger als 512 KB), bei denen die mittleren Latenzen oft im Bereich von mehreren zehn Millisekunden liegen, ist es eine gute Richtlinie, einen OR-Vorgang nach 2 Sekunden erneut zu versuchen. GET PUT Wenn weitere Wiederholungsversuche erforderlich sind, ist ein Backoff die beste Lösung. So empfehlen wir beispielsweise die Ausgabe eines Wiederholungsversuchs nach zwei Sekunden und eines zweiten Versuchs nach weiteren vier Sekunden.

Wenn Ihre Anwendung Anfragen mit fester Größe an Amazon S3 sendet, sollten Sie konsistentere Reaktionszeiten für diese einzelnen Anfragen erwarten. In diesem Fall ist es eine einfache Strategie, das langsamste Prozent der Anforderungen zu identifizieren und diese zu wiederholen. Selbst ein einziger Wiederholungsversuch ist oft erfolgreich für die Reduzierung der Latenz.

Wenn Sie verwenden AWS Key Management Service (AWS KMS) Informationen zur serverseitigen Verschlüsselung finden Sie unter [Grenzwerte](#) in der AWS Key Management Service Im Entwicklerhandbuch finden Sie Informationen zu den Anforderungsraten, die für Ihren Anwendungsfall unterstützt werden.

Horizontale Skalierung und Anforderungsparallelisierung für hohen Durchsatz

Amazon S3 ist ein sehr großes verteiltes System. Um diese Größe zu nutzen, sollten Sie parallele Anforderungen horizontal zu den Amazon S3-Service-Endpunkten skalieren. Zusätzlich zur Verteilung der Anforderungen in Amazon S3 hilft dieses Skalierungskonzept dabei, die Last über mehrere Pfade im Netzwerk zu verteilen.

Für Übertragungen mit hohem Durchsatz empfiehlt Amazon S3 die Verwendung von Anwendungen, die mehrere Verbindungen zu GET oder PUT Daten parallel verwenden. Dies wird beispielsweise von [Amazon S3 Transfer Manager](#) in der AWS Java SDK und die meisten anderen AWS SDKs bieten ähnliche Konstrukte. Für manche Anwendungen können Sie parallele Verbindungen dadurch erreichen, dass Sie Anforderungen gleichzeitig in verschiedenen Anwendungsthreads oder in verschiedenen Anwendungsinstances starten. Das beste Konzept hängt von Ihrer Anwendung und der Struktur der Objekte ab, auf die Sie zugreifen.

Sie können das AWS SDKsPUTAnfragen direkt GET auszustellen, anstatt die Verwaltung von Transfers in der AWS SDK. Mit diesem Ansatz können Sie Ihre Arbeitslast direkter anpassen und gleichzeitig von der SDK Unterstützung für Wiederholungsversuche und der Verarbeitung aller HTTP 503 Antworten profitieren, die auftreten könnten. Wenn Sie große Objekte innerhalb einer Region von Amazon S3 nach [Amazon](#) herunterladen, empfehlen wir generell EC2, gleichzeitige Anfragen für Bytebereiche eines Objekts mit einer Granularität von 8—16 MB zu stellen. Starten Sie eine gleichzeitige Anforderung für jeweils 85-90 MB/s des gewünschten Netzwerkdurchsatzes. Um eine Netzwerkschnittstellenkarte mit 10 Gbit/s (NIC) zu überlasten, könnten Sie etwa 15 gleichzeitige Anfragen über separate Verbindungen verwenden. Sie können die gleichzeitigen Anfragen auf mehr Verbindungen hochskalieren, um sie schneller auszulasten NICs, z. B. 25 Gbit/s oder 100 Gbit/s. NICs

Die Messung der Leistung ist wichtig für die Einstellung der Anzahl der gleichzeitig auszugebenden Anforderungen. Wir empfehlen, mit jeweils einer einzigen Anforderung zu beginnen. Messen Sie die erreichte Netzwerkbandbreite und die Nutzung weiterer Ressourcen durch Ihre Anwendung bei der Verarbeitung der Daten. Sie können dann die Engpassressource (die Ressource mit der höchsten Nutzung) identifizieren und so die Anzahl der Anforderungen ermitteln, die wahrscheinlich nützlich waren. Wenn beispielsweise die Verarbeitung einer Anfrage nach der anderen zu einer CPU Auslastung von 25 Prozent führt, deutet dies darauf hin, dass bis zu vier gleichzeitige Anfragen bearbeitet werden können. Messungen sind sehr wichtig, und es lohnt sich, die Ressourcennutzung zu ermitteln, wenn die Anforderungsrate erhöht wird.

Wenn Ihre Anwendung Anfragen direkt an Amazon S3 sendet, empfehlen wir REST API, einen Pool von HTTP Verbindungen zu verwenden und jede Verbindung für eine Reihe von Anfragen wiederzuverwenden. Durch die Vermeidung einer Verbindungseinrichtung pro Anfrage entfällt die Notwendigkeit, bei jeder Anfrage TCP Slow-Start- und Secure Sockets Layer (SSL) -Handshakes durchzuführen. Informationen zur Verwendung von finden Sie in der [Amazon Simple Storage Service API Reference](#). REST API

Schließlich lohnt es sich, darauf zu achten DNS und zu überprüfen, ob Anfragen über einen großen Pool von Amazon S3 S3-IP-Adressen verteilt werden. DNS Abfragen für Amazon S3 durchlaufen eine große Liste von IP-Endpunkten. Das Caching von Resolvern oder Anwendungscode, der eine einzelne IP-Adresse wiederverwendet, profitiert nicht von der Adressendiversität und dem daraus resultierenden Lastenausgleich. Netzwerk-Utility-Tools wie das `netstat` Befehlszeilentool können die IP-Adressen anzeigen, die für die Kommunikation mit Amazon S3 verwendet werden, und wir stellen Richtlinien für die zu verwendenden DNS Konfigurationen bereit. Weitere Informationen zu diesen Anleitungen finden Sie unter [Senden von Anforderungen](#).

Verwendung von Amazon S3 Transfer Acceleration zur Beschleunigung geographisch disparater Datenübertragungen

[Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration](#) ist effektiv bei der Minimierung oder Beseitigung der durch geographische Entfernung zwischen global verteilten Clients und einer regionalen Anwendung mit Amazon S3 verursachten Latenz. Transfer Acceleration verwendet die weltweit verteilten Edge-Standorte CloudFront für den Datentransport. Das Tool AWS Das Edge-Netzwerk verfügt über Präsenzpunkte an mehr als 50 Standorten. Heute wird es verwendet, um Inhalte über [Amazon Route 53](#) zu verteilen CloudFront und schnelle Antworten auf DNS Anfragen zu geben.

Dazu hilft das Edge-Netzwerk bei der Beschleunigung von Datenübertragungen zu und aus Amazon S3. Es ist ideal für Anwendungen, die Daten über oder zwischen Kontinenten übertragen, eine schnelle Internetverbindung nutzen, große Objekte verwenden oder zahlreiche Inhalte hochladen müssen. Sobald die Daten an einem Edge-Standort eingeht, werden sie über einen optimierten Netzwerkpfad an Ihren Amazon S3-Bucket weitergeleitet. Allgemein gilt: Je weiter Sie von einer Amazon S3-Region entfernt sind, um so größer ist die Geschwindigkeits-Verbesserung, die Sie von Transfer Acceleration erwarten können.

Sie können Transfer Acceleration auf neuen oder bestehenden Buckets einrichten. Sie können einen separaten Amazon S3 Transfer Acceleration Acceleration-Endpunkt verwenden, um AWS Edge-Standorte. Die beste Möglichkeit zur Prüfung, ob Transfer Acceleration die Client-Anfrageleistung erhöht, ist die Verwendung des [Amazon S3-Transfer-Acceleration-Speed-Comparison-Tools](#). Netzwerkkonfigurationen und Bedingungen variieren von Zeit zu Zeit und von Standort zu Standort. Sie werden daher nur für Übertragungen belastet, bei denen Amazon S3 Transfer Acceleration Ihre Upload-Leistung potentiell verbessern kann. Informationen zur Verwendung von Transfer Acceleration mit verschiedenen AWS SDKs, siehe [Aktivieren und Verwenden von S3 Transfer Acceleration](#).

Was ist Amazon S3 on Outposts?

AWS Outposts ist ein vollständig verwalteter Service, der dieselbe AWS Infrastruktur, dieselben AWS Services und Tools für praktisch jedes Rechenzentrum APIs, jeden Colocation-Bereich oder jede lokale Einrichtung bietet und so für ein wirklich konsistentes Hybrid-Erlebnis sorgt. AWS Outposts ist ideal für Workloads, die Zugriff auf lokale Systeme mit niedriger Latenz, lokale Datenverarbeitung, Datenresidenz und Migration von Anwendungen mit lokalen Systemabhängigkeiten erfordern. Weitere Informationen finden Sie unter [Was ist AWS Outposts?](#) im AWS Outposts -Benutzerhandbuch.

Mit Amazon S3 on Outposts können Sie S3-Buckets in Ihren Outposts erstellen und Objekte einfach On-Premises speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse `OUTPOSTS`, die Amazon S3 verwendet APIs und darauf ausgelegt ist, Daten dauerhaft und redundant auf mehreren Geräten und Servern in Ihren Outposts zu speichern. Sie kommunizieren mit Ihrem Outposts-Bucket, indem Sie einen Access Point und eine Endpunktverbindung über eine virtuelle private Cloud (VPC) verwenden.

Sie können für APIs Outposts-Buckets dieselben Funktionen wie für Amazon S3 verwenden, einschließlich Zugriffsrichtlinien, Verschlüsselung und Tagging. Sie können S3 auf Outposts über das AWS Management Console, AWS Command Line Interface (AWS CLI) AWS SDKs, oder REST API verwenden.

- [Funktionsweise von S3 on Outposts](#)
- [Funktionen von S3 on Outposts](#)
- [Zugehörige Services](#)
- [Zugriff auf S3 on Outposts](#)
- [Bezahlung für S3 on Outposts](#)
- [Nächste Schritte](#)

Funktionsweise von S3 on Outposts

S3 on Outposts ist ein Objektspeicherdienst, der Daten als Objekte in Buckets in Ihrem Outpost speichert. Ein Objekt ist eine Datendatei und alle Metadaten, die diese Datei beschreiben. Ein Bucket ist ein Container für Objekte.

Um Ihre Daten in S3 on Outposts zu speichern, müssen Sie zunächst einen Bucket erstellen. Beim Erstellen des Buckets geben Sie einen Bucket-Namen und den Outpost an, der den Bucket enthält. Um auf Ihren S3-on-Outposts-Bucket zuzugreifen und Objektoperationen durchzuführen, erstellen und konfigurieren Sie als Nächstes einen Zugriffspunkt. Sie müssen auch einen Endpunkt erstellen, um Anfragen an Ihren Zugriffspunkt weiterzuleiten.

Access Points vereinfachen den Datenzugriff für jede AWS-Service Kundenanwendung, die Daten in S3 speichert. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind und mit denen Sie Objektvorgänge ausführen können, z. B. `GetObject` und `PutObject`. Jeder Zugriffspunkt verfügt über unterschiedliche Berechtigungen und Netzwerkkontrollen.

Sie können Ihre S3 on Outposts Buckets, Access Points und Endpoints erstellen und verwalten, indem Sie AWS Management Console, AWS CLI, AWS SDKs oder verwenden. REST API Um Objekte in Ihren S3 on Outposts-Bucket hochzuladen und zu verwalten, können Sie das AWS CLI AWS SDKs, oder REST API verwenden.

Regionen

Während der AWS Outposts Bereitstellung erstellen Sie oder AWS erstellen eine Service Link-Verbindung, die Ihren Outpost für Bucket-Operationen und Telemetrie wieder mit der von Ihnen ausgewählten AWS-Region oder Outposts-Heimatregion verbindet. Ein Outpost ist auf Konnektivität zum übergeordneten AWS-Region angewiesen. Das Outposts-Rack ist nicht für getrennte Operationen oder Umgebungen mit eingeschränkter oder keiner Konnektivität ausgelegt. Weitere Informationen finden Sie unter [Outpost-Konnektivität zu AWS-Regionen](#) im AWS Outposts Benutzerhandbuch.

Buckets

Ein Bucket ist ein Behälter für Objekte, die in S3 on Outposts gespeichert werden. Sie können beliebig viele Objekte in einem Bucket speichern und bis zu 100 Buckets pro Konto in einem Outpost haben.

Wenn Sie einen Bucket erstellen, geben Sie einen Bucket-Namen ein und wählen den Outpost, in dem der Bucket angelegt werden soll. Der Name eines erstellten Buckets oder sein Outpost kann nicht nachträglich geändert werden. Bucket-Namen müssen den [Regeln für die Benennung von Amazon-S3-Buckets](#) folgen. In S3 on Outposts sind Bucket-Namen für einen Outpost und eindeutig. `AWS-Kontooutpost-id`, `account-id` und der Bucket-Name müssen die S3-on-Outposts-Buckets identifizieren.

Das folgende Beispiel zeigt das Amazon Resource Name (ARN) -Format für S3 in Outposts-Buckets. Das ARN besteht aus der Region, in der sich Ihr Outpost befindet, Ihrem Outpost-Konto, der Outpost-ID und dem Bucket-Namen.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Access Point ARN oder Access Point-Alias. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN Zugriffspunktformat für S3 auf Outposts, das den `outpost-id` `account-id`, und den Zugriffspunktnamen umfasst:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Weitere Informationen über Buckets finden Sie unter [Arbeiten mit S3-on-Outposts-Buckets](#).

Objekte

Objekte sind die Grundeinheiten, die in S3 on Outposts gespeichert sind. Objekte bestehen aus Objekt- und Metadaten. Metadaten bestehen aus mehreren Name/Wert-Paaren, die das Objekt beschreiben. Diese Paare enthalten einige Standardmetadaten, wie das Datum der letzten Änderung, und HTTP Standardmetadaten wie `Content-Type`. Sie können bei der Speicherung des Objekts auch benutzerdefinierte Metadaten angeben. Ein Objekt wird innerhalb eines Buckets eindeutig durch einen [Schlüssel \(oder Namen\)](#) identifiziert.

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn Sie AWS ein Outpost-Rack installieren, bleiben Ihre Daten lokal in Ihrem Outpost, um die Anforderungen an die Datenresidenz zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da das in der Region gehostet AWS Management Console wird, können Sie die Konsole nicht zum Hochladen oder Verwalten von Objekten in Ihrem Outpost verwenden. Sie können jedoch das, AWS Command Line Interface (AWS CLI) und verwenden RESTAPI, AWS SDKs um Ihre Objekte über Ihre Access Points hochzuladen und zu verwalten.

Schlüssel

Ein Objektschlüssel (oder Schlüsselname) ist der eindeutige Bezeichner für ein Objekt in einem Bucket. Jedes Objekt in einem Bucket besitzt genau einen Schlüssel. Jedes Objekt wird durch die Kombination aus Bucket und Objektschlüssel eindeutig identifiziert.

Das folgende Beispiel zeigt das ARN Format für S3 auf Outposts-Objekten, das den AWS-Region Code für die Region, in der der Outpost beheimatet ist, ID, AWS-Konto Outpost-ID, Bucket-Namen und Objektschlüssel enthält:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

Weitere Informationen über Objektschlüssel finden Sie unter [Arbeiten mit S3-on-Outposts-Objekten](#).

S3-Versioning

Sie können die S3-Versionsverwaltung für Outposts-Buckets verwenden, um mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Mit S3-Versioning können Sie jede Version jedes in Ihren Buckets gespeicherten Objekts beibehalten, abrufen und wiederherstellen. Mit der S3-Versionsverwaltung können Sie Versionen sowohl nach unbeabsichtigten Benutzeraktionen als auch nach Anwendungsfehlern problemlos wiederherstellen.

Weitere Informationen finden Sie unter [Verwalten der S3-Versionierung für Ihren S3-on-Outposts-Bucket](#).

Versions-ID

Wenn Sie die S3-Versionsverwaltung in einem Bucket aktivieren, generiert S3 on Outposts eine eindeutige Versions-ID für jedes Objekt, das dem Bucket hinzugefügt wird. Objekte, die zum Zeitpunkt der Aktivierung des Versioning bereits im Bucket vorhanden waren, haben die Versions-ID null. Wenn Sie diese (oder andere) Objekte beispielsweise mit anderen Operationen ändern, erhalten die neuen Objekte eine eindeutige Versions-ID. [PutObject](#)

Weitere Informationen finden Sie unter [Verwalten der S3-Versionierung für Ihren S3-on-Outposts-Bucket](#).

Speicherklasse und Verschlüsselung

S3 in Outposts bietet eine neue Speicherklasse: S3 Outposts (OUTPOSTS). Die Speicherklasse S3 Outposts ist nur für Objekte verfügbar, die in Buckets auf AWS Outposts gespeichert sind. Wenn Sie versuchen, andere S3-Speicherklassen mit S3 on Outposts zu verwenden, gibt S3 on Outposts den Fehler `InvalidStorageClass` aus.

Standardmäßig werden Objekte, die in der Speicherklasse S3 Outposts (OUTPOSTS) gespeichert sind, mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) verschlüsselt. Weitere Informationen finden Sie unter [Datenverschlüsselung in S3 on Outposts](#).

Bucket-Richtlinie

Eine Bucket-Richtlinie ist eine ressourcenbasierte AWS Identity and Access Management (IAM) Richtlinie, mit der Sie Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte gewähren können. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt.

Bucket-Richtlinien verwenden eine JSON auf IAM Richtlinien basierende Sprache, die überall Standard ist. AWS Sie können Bucket-Richtlinien verwenden, um Berechtigungen für die Objekte in einem Bucket hinzuzufügen oder zu verweigern. Bucket-Richtlinien erlauben oder verweigern Anforderungen basierend auf den Elementen in der Richtlinie. Diese Elemente können den Anforderer, S3-on-Outposts-Aktionen, Ressourcen und Aspekte oder Bedingungen der Anforderung beinhalten (z. B. die IP-Adresse, die für die Anforderung verwendet wird). Sie können beispielsweise eine Bucket-Richtlinie erstellen, die kontoübergreifende Berechtigungen zum Hochladen von Objekten in einen S3-on-Outposts-Bucket gewährt, während gleichzeitig sichergestellt wird, dass der Bucket-Eigentümer die volle Kontrolle über die hochgeladenen Objekte hat. Weitere Informationen finden Sie unter [Beispiele für Amazon S3 S3-Bucket-Richtlinien](#).

In Ihrer Bucket-Richtlinie können Sie Platzhalterzeichen (*) ARNs und andere Werte verwenden, um einer Teilmenge von Objekten Berechtigungen zu erteilen. Sie können beispielsweise den Zugriff auf Gruppen von Objekten steuern, die mit einem gemeinsamen [Präfix](#) beginnen oder mit einer bestimmten Erweiterung wie `.html` enden.

S3-on-Outposts-Zugriffspunkte

S3-on-Outposts-Zugriffspunkte sind benannte Netzwerkkendpunkte mit dedizierten Zugriffsrichtlinien, die beschreiben, wie mit diesem Endpunkt auf Daten zugegriffen werden kann. Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3 on Outposts. Zugriffspunkte sind Buckets zugeordnet, mit denen Sie S3-Objektvorgänge ausführen können, z. B. `GetObject` und `PutObject`.

Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Access Point ARN oder Access Point-Alias. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Jeder Zugriffspunkt verfügt über unterschiedliche Berechtigungen und Netzwerkkontrollen, die S3 on Outposts auf alle Anforderungen anwendet, die über diesen Zugriffspunkt eingehen. Jeder Zugriffspunkt erzwingt eine benutzerdefinierte Zugriffspunktrichtlinie, die in Verbindung mit der Bucket-Richtlinie funktioniert, die dem zugrunde liegenden Bucket zugeordnet ist.

Weitere Informationen finden Sie unter [Zugriff auf Ihre S3-on-Outposts-Buckets und -Objekte](#).

Funktionen von S3 on Outposts

Zugriffsverwaltung

S3 on Outposts bietet Funktionen für die Überwachung und Verwaltung des Zugriffs auf Ihre Buckets und Objekte. Standardmäßig werden S3-on-Outposts-Buckets und -Objekte als privat eingestuft. Sie haben nur Zugriff auf die S3-on-Outposts-Ressourcen, die Sie erstellen.

Um detaillierte Ressourcenberechtigungen zu erteilen, die Ihren speziellen Anwendungsfall unterstützen, oder um die Berechtigungen Ihrer S3-on-Outposts-Ressourcen zu überprüfen, können Sie die folgenden Funktionen verwenden.

- [S3 öffentlichen Zugriff blockieren](#) – Blockieren Sie den öffentlichen Zugriff auf Buckets und Objekte. Für Buckets auf Outposts ist „Öffentlichen Zugriff blockieren“ standardmäßig aktiviert.
- [AWS Identity and Access Management \(IAM\)](#) — IAM ist ein Webservice, mit dem Sie den Zugriff auf AWS Ressourcen, einschließlich Ihrer S3 on Outposts-Ressourcen, sicher kontrollieren können. Mit können Sie Berechtigungen zentral verwalten IAM, mit denen gesteuert wird, auf welche AWS Ressourcen Benutzer zugreifen können. Sie steuern IAM damit, wer authentifiziert (angemeldet) und autorisiert ist (über Berechtigungen verfügt), um Ressourcen zu verwenden.

- [S3-on-Outposts-Zugriffspunkte](#) – Verwalten Sie den Datenzugriff auf freigegebene Datensätze in S3 on Outposts. Zugriffspunkte sind benannte Netzwerkendpunkte mit dedizierten Zugriffsrichtlinien. Zugriffspunkte sind Buckets zugeordnet und können für Objektvorgänge verwendet werden, z. B. GetObject und PutObject.
- [Bucket-Richtlinien](#) — Verwenden Sie eine IAM basierte Richtlinienprache, um ressourcenbasierte Berechtigungen für Ihre S3-Buckets und die darin enthaltenen Objekte zu konfigurieren.
- [AWS Resource Access Manager \(AWS RAM\)](#) — Teilen Sie Ihre S3 on Outposts-Kapazität auf sichere Weise innerhalb Ihrer Organisation oder Ihrer Organisationseinheiten (OUs) in AWS Organizations. AWS-Konten

Speicherprotokollierung und Überwachung

S3 on Outposts bietet Protokollierungs- und Überwachungstools, mit denen Sie überwachen und steuern können, wie Ihre S3-on-Outposts-Ressourcen verwendet werden. Weitere Informationen finden Sie unter [Überwachungstools](#).

- [CloudWatch Amazon-Metriken für S3 on Outposts](#) — Verfolgen Sie den Betriebsstatus Ihrer Ressourcen und machen Sie sich ein Bild von Ihrer Kapazitätsverfügbarkeit.
- [Amazon CloudWatch Events-Ereignisse für S3 on Outposts](#) — Erstellen Sie eine Regel für jedes S3 on API Outposts-Ereignis, um Benachrichtigungen über alle unterstützten CloudWatch Events-Ziele zu erhalten, einschließlich Amazon Simple Queue Service (AmazonSQS), Amazon Simple Notification Service (AmazonSNS) und AWS Lambda
- [AWS CloudTrail Logs für S3 on Outposts](#) — Zeichnet Aktionen auf, die von einem Benutzer, einer Rolle oder einem AWS-Service in S3 auf Outposts ausgeführt wurden. CloudTrail Protokolle bieten Ihnen eine detaillierte API Nachverfolgung von S3-Vorgängen auf Bucket- und Objektebene.

Starke Konsistenz

S3 on Outposts bietet insgesamt eine hohe read-after-write Konsistenz für Objekte PUT und DELETE Anfragen von Objekten in Ihrem S3 on Outposts-Bucket. AWS-Regionen Dieses Verhalten gilt sowohl für Schreibvorgänge neuer Objekte als auch für PUT Anfragen, die bestehende Objekte überschreiben, und für Anfragen. DELETE Darüber hinaus sind Objekt-Tags und Objektmetadaten (z. B. das HEAD Objekt) in S3 on Outposts stark konsistent. Weitere Informationen finden Sie unter [Amazon S3-Datenkonsistenzmodell](#).

Zugehörige Services

Nachdem Sie Daten in S3 on Outposts hochgeladen haben, können Sie sie mit anderen AWS-Services nutzen. Häufig genutzte Services:

- [Amazon Elastic Compute Cloud \(AmazonEC2\)](#) — Bietet sichere und skalierbare Rechenkapazität in der AWS Cloud. Durch die Verwendung von Amazon EC2 müssen Sie nicht im Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwenden EC2, um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten.
- [Amazon Elastic Block Store \(AmazonEBS\) auf Outposts](#) — Verwenden Sie EBS lokale Amazon-Snapshots auf Outposts, um Snapshots von Volumes auf einem Outpost lokal in S3 auf Outposts zu speichern.
- [Amazon Relational Database Service \(AmazonRDS\) auf Outposts](#) — Verwenden Sie RDS lokale Amazon-Backups, um Ihre RDS Amazon-Backups lokal auf Ihrem Outpost zu speichern.
- [AWS DataSync](#) — Automatisieren Sie die Übertragung von Daten zwischen Ihren Outposts und entscheiden Sie AWS-Regionen, was übertragen werden soll, wann übertragen werden soll und wie viel Netzwerkbandbreite verwendet werden soll. S3 on Outposts ist integriert in. AWS DataSync Für On-Premises-Anwendungen, die eine lokale Verarbeitung mit hohem Durchsatz erfordern, bietet S3 on Outposts On-Premises-Objektspeicher, um Datenübertragungen zu minimieren und einen Puffer gegen Netzwerkschwankungen zu bieten, und ermöglicht Ihnen gleichzeitig, Daten einfach zwischen Outposts und AWS-Regionen zu übertragen.

Zugriff auf S3 on Outposts

Sie können mit S3 on Outposts auf eine der folgenden Arten arbeiten:

AWS Management Console

Die Konsole ist eine webbasierte Benutzeroberfläche für die Verwaltung von S3 on Outposts und AWS -Ressourcen. Wenn Sie sich für eine angemeldet haben AWS-Konto, können Sie auf S3 auf Outposts zugreifen, indem Sie sich bei der anmelden AWS Management Console und auf der AWS Management Console Startseite S3 auswählen. Wählen Sie dann Outposts buckets (Outposts-Buckets) aus dem linken Navigationsbereich aus.

AWS Command Line Interface

Sie können die AWS Befehlszeilentools verwenden, um Befehle auszugeben oder Skripte an der Befehlszeile Ihres Systems zu erstellen, um Aufgaben AWS (einschließlich S3) auszuführen.

Das [AWS Command Line Interface \(AWS CLI\)](#) bietet Befehle für eine Vielzahl von AWS-Services. Das AWS CLI wird unter Windows, MacOS und Linux unterstützt. Informationen zu den ersten Schritten finden Sie im [AWS Command Line Interface -Benutzerhandbuch](#). Weitere Informationen zu den Befehlen, die Sie mit S3 on Outposts verwenden können, finden Sie unter [s3api](#), [s3control](#) und [s3outposts](#) in der AWS CLI -Befehlsreferenz.

AWS SDKs

AWS stellt SDKs (Software Development Kits) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen bestehen (Java, Python, Ruby, .NET, iOS, Android usw.). AWS SDKs bieten eine bequeme Möglichkeit, programmatischen Zugriff auf S3 auf Outposts und zu erstellen. AWS Da S3 on Outposts dasselbe verwendet SDKs wie Amazon S3, bietet S3 on Outposts ein einheitliches Erlebnis mit demselben S3 APIs, derselben Automatisierung und denselben Tools.

S3 on Outposts ist ein REST Service. Sie können Anfragen an S3 auf Outposts senden, indem Sie die AWS SDK Bibliotheken verwenden, die die zugrunde liegenden Daten zusammenfassen REST API und Ihre Programmieraufgaben vereinfachen. SDKs kümmern sich beispielsweise um Aufgaben wie das Berechnen von Signaturen, das kryptografische Signieren von Anfragen, das Verwalten von Fehlern und das automatische Wiederholen von Anfragen. Informationen zu den AWS SDKs, einschließlich deren Herunterladen und Installieren, finden Sie unter [Tools](#), auf denen Sie aufbauen können. AWS

Bezahlung für S3 on Outposts

Sie können eine Vielzahl von AWS Outposts Rack-Konfigurationen mit einer Kombination aus EC2 Amazon-Instance-Typen, Amazon EBS Allzweck-Solid-State-Laufwerken (SSD), Volumes (gp2) und S3 on Outposts erwerben. Die Preise beinhalten Lieferung, Installation, Wartung von Infrastruktur-Services sowie Softwarepatches und Upgrades.

Weitere Informationen finden Sie in der [Preisliste für AWS Outposts -Racks](#).

Nächste Schritte

Weitere Informationen zur Arbeit mit S3 on Outposts finden Sie in den folgenden Themen:

- [Einrichten Ihres Outposts](#)
- [Wie unterscheidet sich Amazon S3 on Outposts von Amazon S3?](#)
- [Erste Schritte mit Amazon S3 on Outposts](#)
- [Vernetzung für S3 on Outposts](#)
- [Arbeiten mit S3-on-Outposts-Buckets](#)
- [Arbeiten mit S3-on-Outposts-Objekten](#)
- [Sicherheit in S3 on Outposts](#)
- [Verwaltung von S3-on-Outposts-Speicher](#)
- [Entwickeln mit Amazon S3 on Outposts](#)

Einrichten Ihres Outposts

Um mit Amazon S3 on Outposts zu beginnen, benötigen Sie einen Outpost mit Amazon-S3-Kapazität, der in Ihrer Einrichtung bereitgestellt wird. Weitere Informationen zu den Optionen für die Bestellung eines Outposts und von S3-Kapazitäten finden Sie unter [AWS Outposts](#). Zum Überprüfen, ob Ihre Outposts über S3-Kapazität verfügen, können Sie den API-Aufruf [ListOutpostsWithS3](#) verwenden. Technische Daten und weitere Informationen dazu, wie sich S3 on Outposts von Amazon S3 unterscheidet, finden Sie unter [Wie unterscheidet sich Amazon S3 on Outposts von Amazon S3?](#)

Weitere Informationen finden Sie unter den folgenden Themen.

Themen

- [Einen neuen -Outpost bestellen](#)

Einen neuen -Outpost bestellen

Wenn Sie einen neuen Outpost mit S3-Kapazität bestellen müssen, lesen Sie [Preisliste für AWS Outposts-Racks](#), um die Kapazitätsoptionen für Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS) und Amazon S3 zu verstehen.

Nach der Auswahl der Konfiguration führen Sie die Schritte unter [Erstellen eines Outposts und Bestellen von Outpost-Kapazitäten](#) im AWS Outposts-Benutzerhandbuch aus.

Wie unterscheidet sich Amazon S3 on Outposts von Amazon S3?

Amazon S3 on Outposts stellt Ihrer lokalen AWS Outposts-Umgebung Objektspeicher bereit. Mit S3 on Outposts können Sie die Anforderungen im Hinblick auf lokale Verarbeitung, Datenaufbewahrung und anspruchsvolle Leistung erfüllen, indem Daten in der Nähe von lokalen Anwendungen bleiben. Mithilfe von Amazon-S3-APIs und -Funktionen erleichtert S3 on Outposts das Speichern, Sichern, Markieren, Melden sowie die Kontrolle des Zugriffs auf die Daten in Ihren Outposts. Außerdem wird die AWS-Infrastruktur für Ihre On-Premises-Einrichtung zugunsten eines konsistenten Hybrid-Erlebnisses erweitert.

Weitere Informationen zu den Alleinstellungsmerkmalen von S3 on Outposts finden Sie in den folgenden Themen.

Themen

- [Spezifikationen für S3 auf Outposts](#)
- [Von S3 unterstützte API-Operationen auf Outposts](#)
- [Amazon-S3-Funktionen, die von S3 auf Outposts nicht unterstützt werden](#)
- [Netzwerkanforderungen von S3 on Outposts](#)

Spezifikationen für S3 auf Outposts

- Die maximale Outpost-Bucket-Größe beträgt 50 TB.
- Die maximale Anzahl von Outpost-Buckets beträgt 100 pro AWS-Konto.
- Auf Outpost-Buckets kann nur über Zugriffs- und Endpunkte zugegriffen werden.
- Die maximale Anzahl von Zugriffspunkten pro Outpost-Bucket beträgt 10.
- Zugriffspunkt-Richtlinien sind auf eine Größe von 20 KB beschränkt.
- Der Outpost-Eigentümer kann den Zugriff innerhalb Ihrer Organisation in AWS Organizations mithilfe von AWS Resource Access Manager verwalten. Alle Konten, die Zugriff auf den Außenposten benötigen, müssen sich innerhalb derselben Organisation befinden wie das Eigentümerkonto in AWS Organizations.
- Das S3 in Outposts-Bucket-Eigentümerkonto ist immer der Eigentümer aller Objekte im Bucket.
- Nur das S3 in Outposts-Bucket-Eigentümerkonto kann Vorgänge für den Bucket ausführen.
- Die Objektgrößenbegrenzungen entsprechen denen von Amazon S3.

- Alle auf S3 auf Outposts gespeicherten Objekte werden in der Speicherklasse OUTPOSTS gespeichert.
- Standardmäßig werden alle in der Speicherklasse OUTPOSTS gespeicherten Objekte mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) gespeichert. Sie können auch explizit auswählen, Objekte mithilfe serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) zu speichern.
- Wenn nicht genügend Speicherplatz vorhanden ist, um ein Objekt in Ihrem Outpost zu speichern, gibt die API eine Ausnahme wegen unzureichender Kapazität (ICE) zurück.

Von S3 unterstützte API-Operationen auf Outposts

Eine Liste der von S3 on Outposts unterstützten API-Operationen finden Sie unter [API-Vorgänge in Amazon S3 on Outposts](#).

Amazon-S3-Funktionen, die von S3 auf Outposts nicht unterstützt werden

Mehrere Amazon-S3-Funktionen werden derzeit von Amazon S3 auf Outposts nicht unterstützt. Versuche, sie zu verwenden, werden zurückgewiesen.

- Zugriffskontrolllisten (ACLs)
- Cross-Origin Resource Sharing (CORS)
- S3-BatchVorgänge
- S3-Bestandsberichte
- Ändern der Bucket-Standard-Verschlüsselung
- Öffentliche Buckets
- Multi-Faktor Authentifizierung (MFA) aktivieren
- S3-Lebenszyklusübergänge (abgesehen von der Objektlöschung und dem Abbrechen unvollständiger mehrteiliger Uploads)
- S3-Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen
- Aufrechterhaltung der Objektsperre
- Serverseitige Verschlüsselung mit Schlüsseln, die von AWS Key Management Service (AWS KMS) (SSE-KMS) verwaltet werden
- S3-Replikationszeitkontrolle (S3 RTC)

- Amazon CloudWatch Anfragemetriken
- Metrik-Konfiguration
- Transfer Acceleration
- S3-Ereignis-Benachrichtigungen
- Buckets mit Zahlung durch den Anforderer
- S3 Select
- AWS Lambda-Ereignisse
- Server access logging (Server-Zugriffsprotokollierung)
- HTTP POST-Anforderungen
- SOAP
- Websitezugriff

Netzwerkanforderungen von S3 on Outposts

- Um Anforderungen an einen Zugriffspunkt für S3 in Outposts weiterzuleiten, müssen Sie einen S3-in-Outposts-Endpunkt erstellen und konfigurieren. Für Endpunkte für S3 in Outposts gelten die folgenden Beschränkungen:
 - Jeder Virtual Private Cloud (VPC) in einem Outpost kann ein Endpunkt zugeordnet sein und Sie können bis zu 100 Endpunkte pro Outpost verwenden.
 - Sie können einem Endpunkt mehrere Zugriffspunkte zuordnen.
 - Endpunkte können Sie nur zu VPCs mit CIDR-Blöcken in den Subspaces der folgenden CIDR-Bereiche hinzufügen:
 - 10.0.0.0/8
 - 172.16.0.0/12
 - 192.168.0.0/16
- Endpunkte zu einem Outpost können Sie nur aus VPCs mit nicht überlappenden CIDR-Blöcken erstellen.
- Sie können einen Endpunkt nur aus seinem Outposts-Subnetz erstellen.
- Das Subnetz, das Sie zum Erstellen eines Endpunkts verwenden, muss vier IP-Adressen enthalten, die S3 in Outposts verwenden kann.
- Wenn Sie den kundeneigenen IP-Adresspool (CoIP-Pool) angeben, muss dieser vier IP-Adressen enthalten, die S3 in Outposts verwenden kann.

- Sie können nur einen Endpunkt pro Outpost pro VPC erstellen.

Erste Schritte mit Amazon S3 on Outposts

Mit Amazon S3 in Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden.

Mit Amazon S3 on Outposts können Sie die Amazon-S3-APIs und -Funktionen wie Objektspeicherung, Zugriffsrichtlinien, Verschlüsselung und Tagging auf AWS Outposts wie bei Amazon S3 verwenden. Weitere Informationen zu S3 on Outposts finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Themen

- [Einrichtung IAM mit S3 auf Outposts](#)
- [Erste Schritte unter Verwendung der AWS Management Console](#)
- [Erste Schritte mit dem AWS CLI und SDK for Java](#)

Einrichtung IAM mit S3 auf Outposts

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon S3 on Outposts-Ressourcen zu verwenden. IAM ist eine AWS-Service, die Sie ohne zusätzliche Kosten nutzen können. Standardmäßig haben IAM-Benutzer keine Berechtigungen für S3 auf Outpost-Ressourcen und -Vorgänge. Um Zugriffsberechtigungen für S3 auf Outposts-Ressourcen und API-Operationen IAM zu gewähren, können Sie [Benutzer](#), [Gruppen](#) oder [Rollen](#) erstellen und Berechtigungen anhängen.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Benutzer, IAM die über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Folgen Sie den Anweisungen [unter Erstellen einer Rolle für einen externen Identitätsanbieter \(Federation\)](#) im IAMBenutzerhandbuch.

- IAMBenutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen [unter Eine Rolle für einen IAM Benutzer erstellen](#) im IAMBenutzerhandbuch.
- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Folgen Sie den Anweisungen [unter Hinzufügen von Berechtigungen für einen Benutzer \(Konsole\)](#) im IAMBenutzerhandbuch.

Zusätzlich zu IAM identitätsbasierten Richtlinien unterstützt S3 on Outposts sowohl Bucket- als auch Access-Point-Richtlinien. Bucket-Richtlinien und Zugriffspunkt-Richtlinien sind [ressourcenbasierte Richtlinien](#), die mit der S3-on-Outposts-Ressource verbunden sind.

- Eine Bucket-Richtlinie ist mit dem Bucket verknüpft und erlaubt oder verweigert Anfragen an den Bucket und die darin enthaltenen Objekte auf der Grundlage der Elemente in der Richtlinie.
- Im Gegensatz dazu ist eine Zugriffspunkt-Richtlinie mit dem Zugriffspunkt verbunden und erlaubt oder verweigert Anfragen an den Zugriffspunkt.

Die Zugriffspunkt-Richtlinie funktioniert mit der Bucket-Richtlinie, die dem zugrunde liegenden S3-on-Outposts-Bucket zugeordnet ist. Damit eine Anwendung oder ein Benutzer über einen S3-on-Outposts-Zugriffspunkt auf Objekte in einem S3-on-Outposts-Bucket zugreifen kann, müssen sowohl die Zugriffspunkt- als auch die Bucket-Richtlinie die Anfrage zulassen.

Einschränkungen, die Sie in eine Zugriffspunktrichtlinie einschließen, gelten nur für Anforderungen, die über diesen Zugriffspunkt eingehen. Wenn beispielsweise ein Zugriffspunkt mit einem Bucket verbunden ist, können Sie die Zugriffspunkt-Richtlinie nicht verwenden, um Anfragen, die direkt an den Bucket gerichtet sind, zuzulassen oder zu verweigern. Einschränkungen, die Sie auf eine Bucket-Richtlinie anwenden, können jedoch Anfragen zulassen oder verweigern, die direkt an den Bucket oder über den Zugriffspunkt gestellt werden.

In einer IAM Richtlinie oder einer ressourcenbasierten Richtlinie definieren Sie, welche S3-Aktionen auf Outposts zulässig oder verweigert werden. S3-Aktionen für Outposts entsprechen spezifischen S3-Vorgängen für OutpostsAPI. Aktionen von S3 on Outposts verwenden das Namespace-Präfix `s3-outposts:`. Anfragen an das S3 on Outposts-Steuerelement API in einem AWS-Region und Anfragen an die API Objektendpunkte auf dem Outpost werden mithilfe des Namespace-Präfixes authentifiziert IAM und anhand dieses autorisiert. `s3-outposts:` Um mit S3 auf Outposts zu arbeiten, konfigurieren Sie Ihre IAM Benutzer und autorisieren Sie sie für den `s3-outposts:` IAM Namespace.

Weitere Informationen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 on Outposts](#) in der Service-Autorisierungs-Referenz.

Note

- Zugriffskontrolllisten (ACLs) werden von S3 on Outposts nicht unterstützt.
- S3 on Outposts setzt standardmäßig den Bucket-Besitzer als Objekteigentümer ein, um sicherzustellen, dass der Eigentümer eines Buckets nicht am Zugriff auf oder am Löschen von Objekten gehindert werden kann.
- In S3 on Outposts ist S3 Block Public Access stets aktiviert, um sicherzustellen, dass nie öffentlich auf Objekte zugegriffen werden kann.

Weitere Informationen IAM zur Einrichtung von S3 auf Outposts finden Sie in den folgenden Themen.

Themen

- [Prinzipale für die Richtlinien von S3 on Outposts](#)
- [Ressource ARNs für S3 auf Outposts](#)
- [Beispielrichtlinien für S3 on Outposts](#)
- [Berechtigungen für S3-on-Outposts-Endpunkte](#)
- [Serviceverknüpfte Rollen für S3 on Outposts](#)

Prinzipale für die Richtlinien von S3 on Outposts

Wenn Sie eine ressourcenbasierte Richtlinie erstellen, um Zugriff auf Ihren S3-on-Outposts-Bucket zu gewähren, müssen Sie das `Principal`-Element verwenden, um die Person oder Anwendung

anzugeben, die eine Anfrage für eine Aktion oder einen Vorgang auf dieser Ressource stellen kann. Für S3-on-Outposts-Richtlinien können Sie einen der folgenden Prinzipals verwenden:

- Ein AWS-Konto
- Ein IAM Nutzer
- Eine IAM Rolle
- Alle Prinzipale durch Angabe eines Platzhalters (*) in einer Richtlinie, die ein Condition-Element zur Beschränkung des Zugriffs auf einen bestimmten IP-Bereich verwendet

Important

Sie können keine Richtlinie für einen S3-on-Outposts-Bucket schreiben, die einen Platzhalter (*) im Principal-Element verwendet, es sei denn, die Richtlinie enthält auch eine Condition, die den Zugriff auf einen bestimmten IP-Bereich beschränkt. Mit dieser Beschränkung wird sichergestellt, dass es keinen öffentlichen Zugriff auf Ihren S3-on-Outposts-Bucket gibt. Ein Beispiel finden Sie unter [Beispielrichtlinien für S3 on Outposts](#).

Weitere Informationen zu diesem Principal Element finden Sie unter [AWS JSONPolicy Elements: Principal](#) im IAMBenutzerhandbuch.

Ressource ARNs für S3 auf Outposts

Amazon-Ressourcennamen (ARNs) für S3 auf Outposts enthalten die Outpost-ID zusätzlich zu der AWS-Region, auf die sich der Outpost bezieht, der AWS-Konto ID und dem Ressourcennamen. Um auf Ihre Outposts-Buckets und Objekte zuzugreifen und Aktionen für diese auszuführen, müssen Sie eines der in der folgenden Tabelle aufgeführten ARN Formate verwenden.

Der *partition* Wert in der ARN bezieht sich auf eine Gruppe von AWS-Regionen. Jeder AWS-Konto ist auf eine Partition beschränkt. Im Folgenden werden die unterstützten Partitionen angezeigt:

- aws – AWS-Regionen
- aws-us-gov— Regionen AWS GovCloud (US)

S3 in Outposts-Formaten ARN

Amazon S3 auf Outposts ARN	ARN-Format	Beispiel
Eimer ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> / bucket/ <i>bucket_name</i>	arn:aws:s3-outposts: <i>us-west-2</i> : <i>123456789012</i> :outpost/ <i>op-01ac5d28a6a232904</i> / bucket/ <i>amzn-s3-demo-bucket1</i>
Zugangspunkt ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> /accesspoint/ <i>accesspoint_name</i>	arn:aws:s3-outposts: <i>us-west-2</i> : <i>123456789012</i> :outpost/ <i>op-01ac5d28a6a232904</i> /accesspoint/ <i>access-point-name</i>
Objekt ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> / bucket/ <i>bucket_name</i> / object/ <i>object_key</i>	arn:aws:s3-outposts: <i>us-west-2</i> : <i>123456789012</i> :outpost/ <i>op-01ac5d28a6a232904</i> / bucket/ <i>amzn-s3-demo-bucket1</i> /object/ <i>myobject</i>
Zugriffspunktobjekt S3 on Outposts ARN (wird in Richtlinien verwendet)	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> /accesspoint/ <i>accesspoint_name</i> / object/ <i>object_key</i>	arn:aws:s3-outposts: <i>us-west-2</i> : <i>123456789012</i> :outpost/ <i>op-01ac5d28a6a232904</i> /accesspoint/ <i>access-point-name</i> /object/ <i>myobject</i>

Amazon S3 auf Outposts ARN	ARN-Format	Beispiel
S3 auf Outposts ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i>	arn:aws:s3-outposts: <i>us-west-2</i> : <i>123456789012</i> :outpost/ <i>op-01ac5d28a6a232904</i>

Beispielrichtlinien für S3 on Outposts

Example : Bucket-Richtlinie für S3 auf Outposts mit einem Principal AWS-Konto

Die folgende Bucket-Richtlinie verwendet einen AWS-Konto Principal, um Zugriff auf einen S3 on Outposts-Bucket zu gewähren. Wenn Sie diese Bucket-Richtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version":"2012-10-17",
  "Id":"ExampleBucketPolicy1",
  "Statement":[
    {
      "Sid":"statement1",
      "Effect":"Allow",
      "Principal":{
        "AWS":"123456789012"
      },
      "Action":"s3-outposts:*",
      "Resource":"arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket"
    }
  ]
}
```

Example : S3-on-Outposts-Bucket-Richtlinie mit Platzhalterprinzipal (*) und Bedingungsschlüssel, um den Zugriff auf einen bestimmten IP-Bereich zu beschränken

Die folgende Bucket-Richtlinie verwendet einen Platzhalterprinzipal (*) mit der `aws:SourceIp`-Bedingung, um den Zugriff auf einen bestimmten IP-Bereich zu beschränken. Wenn Sie diese

Bucket-Richtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Id": "ExampleBucketPolicy2",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": { "AWS" : "*" },
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket",
      "Condition" : {
        "IpAddress" : {
          "aws:SourceIp": "192.0.2.0/24"
        },
        "NotIpAddress" : {
          "aws:SourceIp": "198.51.100.0/24"
        }
      }
    }
  ]
}
```

Berechtigungen für S3-on-Outposts-Endpunkte

S3 on Outposts benötigt eigene Berechtigungen für die Verwaltung von S3 on Outposts-Endpunktaktionen. IAM

Note


- Für Endpunkte, die den Zugriffstyp des kundeneigenen IP-Adresspools (CoIP-Pool) verwenden, müssen Sie außerdem über Berechtigungen zum Arbeiten mit IP-Adressen aus Ihrem CoIP-Pool verfügen, wie in der folgenden Tabelle beschrieben.
- Für gemeinsame Konten, die über Outposts auf S3 zugreifen AWS Resource Access Manager, können Benutzer mit diesen gemeinsamen Konten keine eigenen Endpunkte in einem gemeinsamen Subnetz erstellen. Wenn ein Benutzer in einem freigegebenen Konto

seine eigenen Endpunkte verwalten möchte, muss das freigegebene Konto ein eigenes Subnetz in Outposts erstellen. Weitere Informationen finden Sie unter [the section called “Freigabe von S3 on Outposts”](#).

IAMEndpunktbezogene Berechtigungen für S3 on Outposts

Aktion	IAMBerechtigungen
CreateEndpoint	<p>s3-outposts:CreateEndpoint</p> <p>ec2:CreateNetworkInterface</p> <p>ec2:DescribeNetworkInterfaces</p> <p>ec2:DescribeVpcs</p> <p>ec2:DescribeSecurityGroups</p> <p>ec2:DescribeSubnets</p> <p>ec2:CreateTags</p> <p>iam:CreateServiceLinkedRole</p> <p>Für Endpunkte, die den Zugriffstyp des kundeneigenen On-Premises-IP-Adresspools (CoIP-Pool) verwenden, sind die folgenden zusätzlichen Berechtigungen erforderlich:</p> <p>s3-outposts:CreateEndpoint</p> <p>ec2:DescribeCoipPools</p> <p>ec2:GetCoipPoolUsage</p> <p>ec2:AllocateAddress</p> <p>ec2:AssociateAddress</p> <p>ec2:DescribeAddresses</p>

Aktion	IAMBerechtigungen
	ec2:DescribeLocalGatewayRouteTableVpcAssociations
DeleteEndpoint	s3-outposts:DeleteEndpoint ec2:DeleteNetworkInterface ec2:DescribeNetworkInterfaces Für Endpunkte, die den Zugriffstyp des kundeneigenen On-Premises-IP-Adresspools (CoIP-Pool) verwenden, sind die folgenden zusätzlichen Berechtigungen erforderlich: s3-outposts:DeleteEndpoint ec2:DisassociateAddress ec2:DescribeAddresses ec2:ReleaseAddress
ListEndpoints	s3-outposts:ListEndpoints

 Note

Sie können Ressourcen-Tags in einer IAM Richtlinie verwenden, um Berechtigungen zu verwalten.

Serviceverknüpfte Rollen für S3 on Outposts

S3 on Outposts verwendet IAM dienstbezogene Rollen, um einige Netzwerkressourcen in Ihrem Namen zu erstellen. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für S3 on Outposts](#).

Erste Schritte unter Verwendung der AWS Management Console

Mit Amazon S3 in Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Weitere Informationen zu den ersten Schritten mit S3 on Outposts unter Verwendung der Konsole finden Sie in den folgenden Themen. Informationen zu den ersten Schritten unter Verwendung der AWS CLI oder AWS SDK for Java finden Sie unter [Erste Schritte mit dem AWS CLI und SDK for Java](#).

Themen

- [Einen Bucket, einen Zugriffspunkt und einen Endpunkt erstellen](#)
- [Nächste Schritte](#)

Einen Bucket, einen Zugriffspunkt und einen Endpunkt erstellen


Die folgende Vorgehensweise veranschaulicht, wie Sie Ihren ersten Bucket in S3 on Outposts erstellen können. Wenn Sie einen Bucket mit der Konsole erstellen, erstellen Sie auch einen Zugriffspunkt und einen Endpunkt, die mit dem Bucket verknüpft sind, sodass Sie sofort mit dem Speichern von Objekten in Ihrem Bucket beginnen können.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie Outposts-Bucket erstellen.
4. Geben Sie unter Bucket name (Bucket-Name) einen DNS-kompatiblen Namen für Ihren Bucket ein.

Der Bucket-Name muss ...:

- innerhalb des AWS-Konto, des Outposts und der AWS-Region, in der sich der Outpost befindet, eindeutig sein.
- Er muss zwischen 3 und 63 Zeichen lang sein.
- Enthält keine Großbuchstaben.
- mit einem Kleinbuchstaben oder einer Zahl beginnen.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

 **Important**

Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name wird in der URL angezeigt, die auf die Objekte im Bucket verweist.

5. Wählen Sie unter Outpost den Outpost aus, in dem sich der Bucket befinden soll.
6. Legen Sie unter Bucket Versioning (Bucket-Versionsverwaltung) den S3-Versionsverwaltungsstatus für Ihren S3-on-Outposts-Bucket auf eine der folgenden Optionen fest:
 - Disable (Deaktivieren) (Standard) – Der Bucket wird nicht versioniert.
 - Enable (Aktivieren) – Aktiviert die S3-Versionsverwaltung für die Objekte im Bucket. Alle Objekte, die dem Bucket hinzugefügt werden, erhalten eine eindeutige Versions-ID.

Weitere Informationen über das S3-Versioning finden Sie unter [Verwalten der S3-Versionierung für Ihren S3-on-Outposts-Bucket](#).

7. (Optional) Fügen Sie ggf. optional tags (optionale Markierungen) hinzu, die Sie mit dem Outposts-Bucket verknüpfen möchten. Sie können Markierungen nutzen, um Kriterien für einzelne Projekte oder Gruppen von Projekten nachzuverfolgen oder um Ihre Buckets unter Verwendung der Kostenzuordnungs-Markierungen zu kennzeichnen.

Standardmäßig werden alle in Ihrem Outposts-Bucket gespeicherten Objekte mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3)

gespeichert. Sie können auch explizit auswählen, Objekte mithilfe serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) zu speichern. Zum Ändern des Verschlüsselungstyps müssen Sie die REST-API, die AWS Command Line Interface (AWS CLI) oder AWS-SDKs verwenden.

8. Geben Sie im Abschnitt Einstellungen für den Zugriffspunkt für Outposts den Namen des Zugriffspunkts ein.

S3-on-Outposts-Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3 on Outposts. Zugriffspunkte sind benannte Netzwerkendpunkte, die Outposts-Buckets zugeordnet sind, mit denen Sie S3-Objektoperationen ausführen können. Weitere Informationen finden Sie unter [Zugriffspunkte](#).

Zugriffspunkt-Namen müssen innerhalb des Kontos für diese Region und diesen Outpost eindeutig sein und den entsprechen [Einschränkungen von Access Points](#).

9. Wählen Sie die VPC für diesen Amazon-S3-on-Outposts-Zugriffspunkt.

Wenn Sie keine VPC haben, wählen Sie Create VPC (VPC erstellen) aus. Weitere Informationen finden Sie unter [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#).

Eine Virtual Private Cloud (VPC) ermöglicht es Ihnen, AWS-Ressourcen in einem virtuellen Netzwerk zu launchen, das Sie definieren. Dieses virtuelle Netzwerk entspricht weitgehend einem herkömmlichen Netzwerk, wie Sie es in Ihrem Rechenzentrum betreiben würden, kann jedoch die Vorteile der skalierbaren Infrastruktur von nutzen AWS.

10. (Optional für eine vorhandene VPC) Wählen Sie ein Endpoint subnet (Endpoint-Subnetz) für Ihren Endpoint aus.

Ein Subnetz ist ein Bereich an IP-Adressen in Ihrer VPC. Wenn Sie nicht das gewünschte Subnetz haben, wählen Sie Subnetz erstellen. Weitere Informationen finden Sie unter [Vernetzung für S3 on Outposts](#).

11. (Optional für eine vorhandene VPC) Wählen Sie eine Endpoint security group (Endpoint-Sicherheitsgruppe) für Ihren Endpoint aus.

Eine [Sicherheitsgruppe](#) dient als virtuelle Firewall zur Steuerung von ein- und ausgehendem Datenverkehr.

12. (Optional für eine vorhandene VPC) Wählen Sie den Endpoint access type (Endpointzugriffstyp) aus:

- Privat – Zur Verwendung mit der VPC.

- IP im Besitz des Kunden – Zur Verwendung mit einem kundeneigenen IP-Adresspool (CoIP-Pool) Ihres On-Premises-Netzwerks.
13. (Optional) Geben Sie die Outpost access point policy (Outpost-Zugriffspunkt-Richtlinie) an. Die Konsole zeigt automatisch den Amazon-Ressourcennamen (ARN) für den Zugriffspunkt an, den Sie in der Richtlinie verwenden können.
 14. Wählen Sie Outposts-Bucket erstellen.

Note

Es kann bis zu 5 Minuten dauern, bis der Outpost-Endpunkt erstellt und der Bucket einsatzbereit ist. Um zusätzliche Bucket-Einstellungen zu konfigurieren, wählen Sie Details anzeigen.

Nächste Schritte

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn AWS ein Outpost-Rack installiert, bleiben Ihre Daten in Ihrem Outpost lokal, um die Anforderungen an die Datenspeicherorte zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da die AWS Management Console innerhalb der Region gehostet wird, können Sie die Konsole nicht verwenden, um Objekte in Ihrem Outpost hochzuladen oder zu verwalten. Sie können die REST-API, AWS Command Line Interface (AWS CLI) und AWS-SDKs zum Hochladen und Verwalten Ihrer Objekte über Ihre Zugriffspunkte verwenden.

Nachdem Sie einen S3-on-Outposts-Bucket, einen Zugriffspunkt und einen Endpunkt erstellt haben, können Sie die AWS CLI oder das SDK für Java verwenden, um ein Objekt in Ihren Bucket hochzuladen. Weitere Informationen finden Sie unter [Laden Sie ein Objekt in einen S3-Bucket auf Outposts hoch](#).

Erste Schritte mit dem AWS CLI und SDK for Java

Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihren AWS Outposts erstellen und Objekte für Anwendungen, die lokalen Datenzugriff, lokale Datenverarbeitung und Datenresidenz erfordern, einfach vor Ort speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse, S3 Outposts (OUTPOSTS), die die Amazon S3 S3-APIs verwendet und darauf ausgelegt ist, Daten dauerhaft und redundant auf mehreren Geräten und Servern auf Ihrem zu speichern. AWS Outposts Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei Outpost-Buckets dieselben APIs und

Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 auf Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Für die ersten Schritte mit S3 on Outposts müssen Sie einen Bucket, einen Zugriffspunkt und einen Endpunkt erstellen. Anschließend können Sie Ihre Objekte in den Bucket hochladen. Die folgenden Beispiele zeigen Ihnen, wie Sie mit S3 auf Outposts beginnen können, indem Sie das AWS CLI und SDK for Java verwenden. Die ersten Schritte mit der Konsole finden Sie unter [Erste Schritte unter Verwendung der AWS Management Console](#).

Themen

- [Schritt 1: Erstellen eines Buckets](#)
- [Schritt 2: Erstellen eines Zugriffspunkts](#)
- [Schritt 3: Erstellen eines Endpunkts](#)
- [Schritt 4: Hochladen eines Objekts in einen S3-on-Outposts-Bucket](#)

Schritt 1: Erstellen eines Buckets

Im Folgenden AWS CLI und in den SDK-Beispielen für Java wird gezeigt, wie Sie einen S3 on Outposts-Bucket erstellen.

AWS CLI

Example

Im folgenden Beispiel wird ein S3-on-Outposts-Bucket (`s3-outposts:CreateBucket`) mithilfe der AWS CLI erstellt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-id op-01ac5d28a6a232904
```

SDK for Java

Example

Im folgenden Beispiel wird ein S3-on-Outposts-Bucket (`s3-outposts:CreateBucket`) mit dem SDK für Java erstellt.

```
import com.amazonaws.services.s3control.model.*;

public String createBucket(String bucketName) {

    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()
        .withBucket(bucketName)
        .withOutpostId(OutpostId)
        .withCreateBucketConfiguration(new CreateBucketConfiguration());

    CreateBucketResult respCreateBucket =
s3ControlClient.createBucket(reqCreateBucket);
    System.out.printf("CreateBucket Response: %s%n", respCreateBucket.toString());

    return respCreateBucket.getBucketArn();

}
```

Schritt 2: Erstellen eines Zugriffspunkts

Um auf Ihren Amazon-S3-on-Outposts-Bucket zuzugreifen, müssen Sie einen Zugriffspunkt erstellen und konfigurieren. In diesen Beispielen erfahren Sie, wie Sie mit dem AWS CLI und dem SDK for Java einen Access Point erstellen.

Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie Amazon S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Access Points unterstützen nur virtual-host-style Adressierung.

AWS CLI

Example

Im folgenden AWS CLI Beispiel wird ein Access Point für einen Outposts-Bucket erstellt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-access-point --account-id 123456789012
--name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

SDK for Java

Example

Im folgenden Beispiel für SDK für Java wird ein Zugriffspunkt für einen Outposts-Bucket erstellt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.services.s3control.model.*;

public String createAccessPoint(String bucketArn, String accessPointName) {

    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withName(accessPointName)
        .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP =
s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s%n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();
}
```

Schritt 3: Erstellen eines Endpunkts

Wenn Sie Anforderungen an einen Zugriffspunkt für Amazon S3 on Outposts weiterleiten möchten, müssen Sie einen S3-on-Outposts-Endpunkt erstellen und konfigurieren. Zum Erstellen eines Endpunkts muss Ihr Service-Link eine aktive Verbindung mit Ihrer Outposts-Heimatregion aufweisen. Jeder Virtual Private Cloud (VPC) in Ihrem Outpost kann ein Endpunkt zugeordnet sein. Weitere Informationen zu Endpunktkontingenten finden Sie unter [Netzwerkanforderungen von S3 on Outposts](#). Sie müssen einen Endpunkt erstellen, um auf Ihre Outposts-Buckets zugreifen und Objekt-Operationen ausführen zu können. Weitere Informationen finden Sie unter [Endpunkte](#).

Diese Beispiele zeigen Ihnen, wie Sie mit dem AWS CLI und dem SDK for Java einen Endpunkt erstellen. Weitere Informationen zu den erforderlichen Berechtigungen für das Erstellen und Verwalten von Endpunkten finden Sie unter [Berechtigungen für S3-on-Outposts-Endpunkte](#).

AWS CLI

Example

Im folgenden AWS CLI Beispiel wird mithilfe des VPC-Ressourcenzugriffstyps ein Endpunkt für einen Outpost erstellt. Die VPC ist vom Subnetz abgeleitet. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

Im folgenden AWS CLI Beispiel wird mithilfe des Zugriffstyps des kundeneigenen IP-Adresspools (CoIP-Pool) ein Endpunkt für einen Outpost erstellt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

SDK for Java

Example

Im folgenden SDK für Java-Beispiel wird ein Endpunkt für einen Outpost erstellt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;

public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
        .withOutpostId("op-0d79779cef3c30a40")
        .withSubnetId("subnet-8c7a57c5")
        .withSecurityGroupId("sg-ab19e0d1")
        .withAccessType("CustomerOwnedIp")
        .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
```

```
// Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type
is
// customer-owned IP address pool (CoIP pool)
CreateEndpointResult createEndpointResult =
s3OutpostsClient.createEndpoint(createEndpointRequest);
System.out.println("Endpoint is created and its ARN is " +
createEndpointResult.getEndpointArn());
}
```

Schritt 4: Hochladen eines Objekts in einen S3-on-Outposts-Bucket

Informationen zum Hochladen eines Objekts finden Sie unter [Laden Sie ein Objekt in einen S3-Bucket auf Outposts hoch](#).

Vernetzung für S3 on Outposts

Sie können Amazon S3 on Outposts verwenden, um Objekte lokal für Anwendungen zu speichern und abzurufen, die lokalen Datenzugriff, Datenverarbeitung und Datenresidenz erfordern. In diesem Abschnitt werden die Netzwerkanforderungen für den Zugriff auf S3 on Outposts beschrieben.

Themen

- [Auswählen des Netzwerkzugriffstyps](#)
- [Zugriff auf Ihre S3-on-Outposts-Buckets und -Objekte](#)
- [Kontenübergreifende Elastic-Network-Schnittstellen](#)

Auswählen des Netzwerkzugriffstyps

Sie können von einer VPC oder von Ihrem lokalen Netzwerk aus auf S3 on Outposts zugreifen. Sie kommunizieren mit Ihrem Outposts-Bucket über einen Zugriffspunkt und eine Endpunktverbindung. Dadurch bleibt der Datenverkehr zwischen Ihrer VPC und Ihren S3-on-Outposts-Buckets innerhalb des AWS-Netzwerks. Beim Erstellen eines Endpunkts müssen Sie den Endpunktzugriffstyp entweder als `Private` (für VPC-Routing) oder `CustomerOwnedIp` (für einen kundeneigenen IP-Adresspool [CoIP-Pool]) angeben.

- `Private` (für VPC-Routing) – Wenn Sie den Zugriffstyp nicht angeben, verwendet S3 on Outposts standardmäßig `Private`. Mit dem Zugriffstyp `Private` benötigen Instances in Ihrer VPC keine öffentlichen IP-Adressen, um mit Ressourcen in Ihrem Outpost zu kommunizieren. Sie können von

einer VPC aus mit S3 on Outposts arbeiten. Der Zugriff auf diesen Endpunkttyp ist über direktes VPC-Routing über Ihr lokales Netzwerk möglich. Weitere Informationen finden Sie unter [Roouting-Tabellen für lokale Gateways](#) im AWS-Outposts-Benutzerhandbuch.

- `CustomerOwnedIp`(für CoIP-Pool) – Wenn Sie den Zugriffstyp `Private` nicht standardmäßig verwenden und `CustomerOwnedIp` auswählen, müssen Sie einen IP-Adressbereich angeben. Sie können diesen Zugriffstyp verwenden, um mit S3 on Outposts sowohl aus Ihrem On-Premises-Netzwerk als auch innerhalb einer VPC zu arbeiten. Wenn Sie auf S3 on Outposts innerhalb einer VPC zugreifen, ist Ihr Datenverkehr auf die Bandbreite des lokalen Gateways beschränkt.

Zugriff auf Ihre S3-on-Outposts-Buckets und -Objekte

Um auf Ihre S3 on Outposts-Buckets und -Objekte zugreifen zu können, benötigen Sie Folgendes:

- Ein Zugriffspunkt für die VPC.
- Ein Endpunkt für die gleiche VPC.
- Eine aktive Verbindung zwischen Ihrem Outpost und Ihrer AWS-Region. Weitere Informationen über die Verbindungserstellung Ihres Outposts zu einer Region finden Sie unter [Outpost-Konnektivität zu AWS-Regionen](#) im AWS-Outposts-Benutzerhandbuch.

Weitere Informationen zum Zugriff auf Buckets und Objekte in S3 on Outposts finden Sie unter [Arbeiten mit S3-on-Outposts-Buckets](#) und [Arbeiten mit S3-on-Outposts-Objekten](#).

Kontenübergreifende Elastic-Network-Schnittstellen

S3-on-Outposts-Endpunkte sind benannte Ressourcen mit Amazon-Ressourcennamen (ARNs). Wenn diese Endpunkte erstellt werden, richtet AWS Outposts mehrere kontoübergreifende Elastic-Network-Schnittstellen ein. Die kontoübergreifenden Elastic-Network-Schnittstellen von S3 on Outposts sind wie andere Netzwerkschnittstellen mit einer Ausnahme: S3 on Outposts ordnet die kontoübergreifenden Elastic-Network-Schnittstellen Amazon-EC2-Instances zu.

Das S3-on-Outposts-Domain-Name-System (DNS) verteilt Ihre Anfragen über die kontoübergreifende Elastic-Network-Schnittstelle. S3 on Outposts erstellt die kontoübergreifende Elastic Network-Schnittstelle in Ihrem AWS-Konto, die im Bereich Netzwerkschnittstellen der Amazon EC2-Konsole sichtbar ist.

Für Endpunkte, die den CoIP-Pool-Zugriffstyp verwenden, weist S3 on Outposts IP-Adressen der kontoübergreifenden Elastic-Network-Schnittstelle aus dem konfigurierten CoIP-Pool zu und ordnet sie dieser zu.

Arbeiten mit S3-on-Outposts-Buckets

Mit Amazon S3 on Outposts können Sie S3-Buckets in Ihren AWS Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premises speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie können bei Outpost-Buckets dieselben APIs und Funktionen wie in Amazon S3 verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Sie kommunizieren mit Ihrenm Outpost-Buckets über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Für den Zugriff auf Ihre S3-on-Outposts-Buckets und -Objekte benötigen Sie einen Zugriffspunkt für die VPC und einen Endpunkt für dieselbe VPC. Weitere Informationen finden Sie unter [Vernetzung für S3 on Outposts](#).

Buckets

In S3 on Outposts sind Bucket-Namen für einen Outpost eindeutig und erfordern AWS-Region-Code für die Region, in der sich der Outpost befindet, die AWS-Konto-ID, die Outpost ID und den Bucket-Namen, um sie zu identifizieren.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Weitere Informationen finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Zugriffspunkte

Amazon S3 on Outposts unterstützt reine Virtual-Private-Cloud(VPC)-Zugriffspunkte als einzige Möglichkeit, auf Ihre Outposts-Buckets zuzugreifen.

Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie Amazon S3-Objekt-Vorgänge ausführen können, z. B. GetObject

und `PutObject`. Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Zugriffspunkte unterstützen ausschließlich die virtuelle, Host-artige Adressierung.

Im folgenden Beispiel wird das ARN-Format gezeigt, das Sie für S3-on-Outposts-Zugriffspunkte verwenden. Der Zugriffspunkt-ARN umfasst den AWS-Region-Code für die Region, in der sich der Outpost befindet, die AWS-Konto-ID, die Outpost-ID und den Namen des Zugriffspunkts.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Endpunkte

Um Anforderungen an einen Zugriffspunkt für S3 on Outposts weiterzuleiten, müssen Sie einen S3-on-Outposts-Endpunkt erstellen und konfigurieren. Mit S3-on-Outposts-Endpunkten können Sie Ihre VPC privat mit Ihrem Outpost-Bucket verbinden. S3-on-Outposts-Endpunkte sind virtuelle Uniform Resource Identifiers (URIs) des Einstiegspunkts zu Ihrem S3-on-Outposts-Bucket. Es handelt sich bei ihnen um horizontal skalierte, redundante und hochverfügbare VPC-Komponenten.

Jeder Virtual Private Cloud (VPC) in Ihrem Outpost kann ein Endpunkt zugeordnet sein und Sie können bis zu 100 Endpunkte pro Outpost verwenden. Sie müssen diese Endpunkte erstellen, um auf Ihren Outpost-Bucket zuzugreifen und Objektvorgänge ausführen zu können. Das Erstellen dieser Endpunkte ermöglicht auch, dass das API-Modell und das Verhalten identisch sind, indem dieselben Vorgänge in S3 und S3 on Outposts ausgeführt werden.

API-Vorgänge in S3 on Outposts

Um Outpost-Bucket-API-Vorgänge zu verwalten, hostet S3 on Outposts einen separaten Endpunkt, der sich vom Amazon-S3-Endpunkt unterscheidet. Dieser Endpunkt ist `s3-outposts.region.amazonaws.com`.

Um dieselben Amazon-S3-API-Vorgänge zu verwenden, müssen Sie den Bucket und die Objekte im korrekten ARN-Format signieren. Sie müssen ARNs an API-Vorgänge übergeben, damit Amazon S3 feststellen kann, ob die Anfrage für Amazon S3 (`s3-control.region.amazonaws.com`) oder S3 on Outposts (`s3-outposts.region.amazonaws.com`) gilt. Basierend auf dem ARN-Format kann S3 die Anfrage dann entsprechend signieren und weiterleiten.

Wenn die Anforderung an die Amazon S3-Steuerebene gesendet wird, extrahiert das SDK die Komponenten aus dem ARN und fügt einen zusätzlichen Header `x-amz-outpost-id` mit dem Wert

outpost-id ein, der aus dem ARN extrahiert wurde. Der Service-Name aus dem ARN wird für die Signierung der Anforderung verwendet, bevor sie an den S3-on-Outposts-Endpunkt weitergeleitet wird. Dieses Verhalten gilt für alle API-Vorgänge, die vom `s3control`-Client verarbeitet werden.

In der folgenden Tabelle sind die fortschrittlichen API-Vorgänge für Amazon S3 on Outposts und ihre Änderungen im Verhältnis zu Amazon S3 aufgeführt.

API	S3-on-Outposts-Parameterwert
CreateBucket	Bucket-Name wie ARN, Outpost-ID
ListRegionalBuckets	Outpost-ID
DeleteBucket	Bucket-Name als ARN
DeleteBucketLifecycleConfiguration	Bucket-Name als ARN
GetBucketLifecycleConfiguration	Bucket-Name als ARN
PutBucketLifecycleConfiguration	Bucket-Name als ARN
GetBucketPolicy	Bucket-Name als ARN
PutBucketPolicy	Bucket-Name als ARN
DeleteBucketPolicy	Bucket-Name als ARN
GetBucketTagging	Bucket-Name als ARN
PutBucketTagging	Bucket-Name als ARN
DeleteBucketTagging	Bucket-Name als ARN
CreateAccessPoint	Name des Zugriffspunkts als ARN
DeleteAccessPoint	Name des Zugriffspunkts als ARN
GetAccessPoint	Name des Zugriffspunkts als ARN

API	S3-on-Outposts-Parameterwert
GetAccessPoint	Name des Zugriffspunkts als ARN
ListAccessPoints	Name des Zugriffspunkts als ARN
PutAccessPointPolicy	Name des Zugriffspunkts als ARN
GetAccessPointPolicy	Name des Zugriffspunkts als ARN
DeleteAccessPointPolicy	Name des Zugriffspunkts als ARN

Erstellen und Verwalten von S3 on Outposts-Buckets

Weitere Informationen zum Erstellen und Verwalten von S3-on-Outposts-Buckets finden Sie in den folgenden Themen.

Themen

- [Erstellen eines S3-on-Outposts-Buckets](#)
- [Hinzufügen von Tags für S3-on-Outposts-Buckets](#)
- [Verwalten des Zugriffs auf einen Amazon-S3-on-Outposts-Bucket mit einer Bucket-Richtlinie](#)
- [Auflisten von Amazon-S3-on-Outposts-Buckets](#)
- [Abrufen eines S3-on-Outposts-Buckets mithilfe der AWS CLI und des SDK for Java](#)
- [Löschen Ihres Amazon-S3-on-Outposts-Buckets](#)
- [Arbeiten mit Zugriffspunkten von Amazon S3 on Outposts](#)
- [Arbeiten mit Amazon-S3-on-Outposts-Endpunkten](#)

Erstellen eines S3-on-Outposts-Buckets

Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei

Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Note

Das AWS-Konto, das den Bucket erstellt, besitzt ihn und ist das einzige, das ihm Aktionen zuweisen kann. Buckets verfügen über Konfigurationseigenschaften wie Outpost, Tags, Standard-Verschlüsselung und Zugriffspunkteinstellungen. Zu den Zugriffspunkteinstellungen gehören die Virtual Private Cloud (VPC), die Zugriffspunkt-Richtlinie für den Zugriff auf die Objekte im Bucket sowie andere Metadaten. Weitere Informationen finden Sie unter [Spezifikationen für S3 auf Outposts](#).

Wenn Sie einen Bucket erstellen möchten, der AWS PrivateLink verwendet, um in Ihrer Virtual Private Cloud (VPC) über Schnittstellen-VPC-Endpunkte Zugriff auf die Bucket- und Endpunkt-Verwaltung bereitzustellen, rufen Sie [AWS PrivateLink für S3 auf Outposts](#) auf.

Die folgenden Beispiele zeigen, wie Sie einen S3-on-Outposts-Bucket mithilfe der AWS Management Console, der AWS Command Line Interface (AWS CLI) und AWS SDK for Java erstellen.


Verwenden der S3-Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie Outposts-Bucket erstellen.
4. Geben Sie unter Bucket name (Bucket-Name) einen DNS-kompatiblen Namen für Ihren Bucket ein.

Der Bucket-Name ...:

- innerhalb des AWS-Konto, des Outposts und der AWS-Region, in der sich der Outpost befindet, eindeutig sein.
- Er muss zwischen 3 und 63 Zeichen lang sein.
- Enthält keine Großbuchstaben.
- mit einem Kleinbuchstaben oder einer Zahl beginnen.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter [Regeln für die Benennung von Buckets](#).

 **Important**

Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name wird in der URL angezeigt, die auf die Objekte im Bucket verweist.

5. Wählen Sie unter Outpost den Outpost aus, in dem sich der Bucket befinden soll.
6. Legen Sie unter Bucket Versioning (Bucket-Versionsverwaltung) den S3-Versionsverwaltungsstatus für Ihren S3-on-Outposts-Bucket auf eine der folgenden Optionen fest:
 - Disable (Deaktivieren) (Standard) – Der Bucket wird nicht versioniert.
 - Enable (Aktivieren) – Aktiviert die S3-Versionsverwaltung für die Objekte im Bucket. Alle Objekte, die dem Bucket hinzugefügt werden, erhalten eine eindeutige Versions-ID.

Weitere Informationen über das S3-Versioning finden Sie unter [Verwalten der S3-Versionierung für Ihren S3-on-Outposts-Bucket](#).

7. (Optional) Fügen Sie ggf. optional tags (optionale Markierungen) hinzu, die Sie mit dem Outposts-Bucket verknüpfen möchten. Sie können Markierungen nutzen, um Kriterien für einzelne Projekte oder Gruppen von Projekten nachzuverfolgen oder um Ihre Buckets unter Verwendung der Kostenzuordnungs-Markierungen zu kennzeichnen.

Standardmäßig werden alle in Ihrem Outposts-Bucket gespeicherten Objekte mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) gespeichert. Sie können auch explizit auswählen, Objekte mithilfe serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) zu speichern. Zum Ändern des Verschlüsselungstyps müssen Sie die REST-API, die AWS Command Line Interface (AWS CLI) oder AWS-SDKs verwenden.

8. Geben Sie im Abschnitt Einstellungen für den Zugriffspunkt für Outposts den Namen des Zugriffspunkts ein.

S3-on-Outposts-Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3 on Outposts. Zugriffspunkte sind benannte Netzwerkendpunkte, die Outposts-Buckets zugeordnet sind, mit denen Sie S3-Objektoperationen ausführen können. Weitere Informationen finden Sie unter [Zugriffspunkte](#).

Zugriffspunkt-Namen müssen innerhalb des Kontos für diese Region und diesen Outpost eindeutig sein und den entsprechen [Einschränkungen von Access Points](#).

9. Wählen Sie die VPC für diesen Amazon-S3-on-Outposts-Zugriffspunkt.

Wenn Sie keine VPC haben, wählen Sie Create VPC (VPC erstellen) aus. Weitere Informationen finden Sie unter [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#).

Eine Virtual Private Cloud (VPC) ermöglicht es Ihnen, AWS-Ressourcen in einem virtuellen Netzwerk zu launchen, das Sie definieren. Dieses virtuelle Netzwerk entspricht weitgehend einem herkömmlichen Netzwerk, wie Sie es in Ihrem Rechenzentrum betreiben würden, kann jedoch die Vorteile der skalierbaren Infrastruktur von AWS nutzen.

10. (Optional für eine vorhandene VPC) Wählen Sie ein Endpoint subnet (Endpunkt-Subnetz) für Ihren Endpunkt aus.

Ein Subnetz ist ein Bereich an IP-Adressen in Ihrer VPC. Wenn Sie nicht das gewünschte Subnetz haben, wählen Sie Subnetz erstellen. Weitere Informationen finden Sie unter [Vernetzung für S3 on Outposts](#).

11. (Optional für eine vorhandene VPC) Wählen Sie eine Endpoint security group (Endpunkt-Sicherheitsgruppe) für Ihren Endpunkt aus.

Eine [Sicherheitsgruppe](#) dient als virtuelle Firewall zur Steuerung von ein- und ausgehendem Datenverkehr.

12. (Optional für eine vorhandene VPC) Wählen Sie den Endpoint access type (Endpunktzugriffstyp) aus:

- Privat – Zur Verwendung mit der VPC.
- IP im Besitz des Kunden – Zur Verwendung mit einem kundeneigenen IP-Adresspool (CoIP-Pool) Ihres On-Premises-Netzwerks.

13. (Optional) Geben Sie die Outpost access point policy (Outpost-Zugriffspunkt-Richtlinie) an. Die Konsole zeigt automatisch den Amazon-Ressourcennamen (ARN) für den Zugriffspunkt an, den Sie in der Richtlinie verwenden können.

14. Wählen Sie Outposts-Bucket erstellen.

Note

Es kann bis zu 5 Minuten dauern, bis der Outpost-Endpunkt erstellt und der Bucket einsatzbereit ist. Um zusätzliche Bucket-Einstellungen zu konfigurieren, wählen Sie Details anzeigen.

Verwendung von AWS CLI

Example

Im folgenden Beispiel wird ein S3-on-Outposts-Bucket (`s3-outposts:CreateBucket`) mithilfe der AWS CLI erstellt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-id op-01ac5d28a6a232904
```

Verwenden des AWS-SDKs für Java

Example

Im folgenden Beispiel wird ein S3-on-Outposts-Bucket (`s3-outposts:CreateBucket`) mit dem SDK für Java erstellt.

```
import com.amazonaws.services.s3control.model.*;

public String createBucket(String bucketName) {

    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()
        .withBucket(bucketName)
        .withOutpostId(OutpostId)
        .withCreateBucketConfiguration(new CreateBucketConfiguration());

    CreateBucketResult respCreateBucket =
        s3ControlClient.createBucket(reqCreateBucket);
    System.out.printf("CreateBucket Response: %s\n", respCreateBucket.toString());

    return respCreateBucket.getBucketArn();
}
```

}

Hinzufügen von Tags für S3-on-Outposts-Buckets

Sie können Tags für Ihre Amazon-S3-on-Outposts-Buckets hinzufügen, um die Speicherkosten oder andere Kriterien für einzelne Projekte oder Gruppen von Projekten zu verfolgen.

Note

Das AWS-Konto, das den Bucket erstellt, besitzt ihn und ist das einzige, das seine Markierungen ändern kann.

Verwenden der S3-Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, dessen Tags Sie bearbeiten möchten.
4. Wählen Sie die Registerkarte Properties (Eigenschaften) aus.
5. Wählen Sie unter Tags, die Option Edit (Bearbeiten) aus.
6. (Optional) Wählen Sie für jede Markierung Add new tag (Neue Markierung hinzufügen) und geben Sie den Schlüsselnamen unter Key (Schlüssel) und den Wert unter Value (Wert) ein.

Fügen Sie alle Tags hinzu, die Sie mit einem Outposts-Bucket verknüpfen möchten, um andere Kriterien für einzelne Projekte oder Gruppen von Projekten zu verfolgen.

7. Wählen Sie Save Changes (Änderungen speichern).

Verwendung der AWS CLI

Das folgende AWS CLI-Beispiel wendet eine Markierungskonfiguration auf einen S3-on-Outposts-Bucket an, wobei ein JSON-Dokument im aktuellen Ordner verwendet wird, das Tags (*tagging.json*) angibt. Zum Verwenden dieses Beispiels ersetzen Sie jeden *user input placeholder* durch Ihre eigenen Informationen.

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --tagging file://tagging.json
```

tagging.json

```
{
  "TagSet": [
    {
      "Key": "organization",
      "Value": "marketing"
    }
  ]
}
```

Das folgende AWS CLI-Beispiel wendet eine Markierungskonfiguration direkt von der Befehlszeile aus auf einen S3-on-Outposts-Bucket an.

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --tagging 'TagSet=[{Key=organization,Value=marketing}]'
```

Weitere Informationen über diesen Befehl finden Sie unter [put-bucket-tagging](#) in der AWS CLI-Referenz.

Verwalten des Zugriffs auf einen Amazon-S3-on-Outposts-Bucket mit einer Bucket-Richtlinie

Eine Bucket-Richtlinie ist eine auf Ressourcen basierende AWS Identity and Access Management (IAM)-Richtlinie, die Sie verwenden können, um Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte zu erteilen. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt. Weitere Informationen finden Sie unter [Bucket-Richtlinie](#).

Sie können Ihre Bucket-Richtlinie aktualisieren, um den Zugriff auf Ihren Amazon-S3-on-Outposts-Bucket zu verwalten. Weitere Informationen finden Sie unter den folgenden Themen.

Themen

- [Hinzufügen oder Bearbeiten einer Bucket-Richtlinie für einen Amazon-S3-on-Outposts-Bucket](#)
- [Anzeigen der Bucket-Richtlinie für Ihren Amazon-S3-on-Outposts-Bucket](#)
- [Löschen der Bucket-Richtlinie für Ihren Amazon-S3-on-Outposts-Bucket](#)

- [Beispiele für Bucket-Richtlinien](#)

Hinzufügen oder Bearbeiten einer Bucket-Richtlinie für einen Amazon-S3-on-Outposts-Bucket

Eine Bucket-Richtlinie ist eine auf Ressourcen basierende AWS Identity and Access Management (IAM)-Richtlinie, die Sie verwenden können, um Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte zu erteilen. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt. Weitere Informationen finden Sie unter [Bucket-Richtlinie](#).

In den folgenden Themen erfahren Sie, wie Sie Ihre Bucket-Richtlinie für Amazon S3 on Outposts mithilfe der AWS Management Console, der AWS Command Line Interface (AWS CLI) oder AWS SDK for Java aktualisieren.

Verwenden der S3-Konsole

Eine Bucket-Richtlinie erstellen oder bearbeiten

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, dessen Bucket-Richtlinie Sie bearbeiten möchten.
4. Wählen Sie die Registerkarte Permissions (Berechtigungen).
5. Wählen Sie im Abschnitt Outposts bucket policy (Outposts-Bucket-Richtlinie) die Option Edit (Bearbeiten) aus, um eine neue Richtlinie zu erstellen oder zu bearbeiten.

Sie können nun die S3-on-Outposts-Bucket-Richtlinie hinzufügen oder bearbeiten. Weitere Informationen finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

Verwendung der AWS CLI

Im folgenden AWS CLI-Beispiel wird eine Richtlinie für einen Outposts-Bucket eingerichtet.

1. Speichern Sie die folgende Bucket-Richtlinie in einer JSON-Datei. In diesem Beispiel heißt die Datei `policy1.json`. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
{
  "Version":"2012-10-17",
  "Id":"testBucketPolicy",
  "Statement":[
    {
      "Sid":"st1",
      "Effect":"Allow",
      "Principal":{"
        "AWS":"123456789012"
      }},
      "Action":"s3-outposts:*",
      "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket"
    }
  ]
}
```

2. Senden Sie die JSON-Datei als Teil des CLI-Befehls `put-bucket-policy`. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control put-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --policy file://policy1.json
```

Verwenden des AWS-SDKs für Java

Im folgenden SDK für Java-Beispiel wird eine Richtlinie für einen Outposts-Bucket gesetzt.

```
import com.amazonaws.services.s3control.model.*;

public void putBucketPolicy(String bucketArn) {

    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testBucketPolicy\",
    \"Statement\": [{\"Sid\":\"st1\", \"Effect\":\"Allow\", \"Principal\": {\"AWS\": \"\" +
    AccountId+ \"\"}, \"Action\": \"s3-outposts:*\", \"Resource\": \"\" + bucketArn + \"\"}]}";

    PutBucketPolicyRequest reqPutBucketPolicy = new PutBucketPolicyRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withPolicy(policy);
```

```
PutBucketPolicyResult respPutBucketPolicy =
s3ControlClient.putBucketPolicy(reqPutBucketPolicy);
System.out.printf("PutBucketPolicy Response: %s%n",
respPutBucketPolicy.toString());
}
```

Anzeigen der Bucket-Richtlinie für Ihren Amazon-S3-on-Outposts-Bucket

Eine Bucket-Richtlinie ist eine auf Ressourcen basierende AWS Identity and Access Management (IAM)-Richtlinie, die Sie verwenden können, um Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte zu erteilen. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt. Weitere Informationen finden Sie unter [Bucket-Richtlinie](#).

In den folgenden Themen erfahren Sie, wie Sie Ihre Bucket-Richtlinie für Amazon S3 on Outposts mithilfe der AWS Management Console, der AWS Command Line Interface (AWS CLI) oder AWS SDK for Java anzeigen.

Verwenden der S3-Konsole

Eine Bucket-Richtlinie erstellen oder bearbeiten

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, dessen Berechtigung Sie bearbeiten möchten.
4. Wählen Sie die Registerkarte Permissions (Berechtigungen) aus.
5. Im Abschnitt Outposts bucket policy (Outposts-Bucket-Richtlinie) können Sie Ihre vorhandene Bucket-Richtlinie überprüfen. Weitere Informationen finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

Verwendung der AWS CLI

Im folgenden AWS CLI-Beispiel wird eine Richtlinie für einen Outposts-Bucket erhalten. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.


```
aws s3control get-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-  
bucket
```

Verwenden des AWS-SDKs für Java

Im folgenden SDK für Java-Beispiel wird eine Richtlinie für einen Outposts-Bucket abgerufen.

```
import com.amazonaws.services.s3control.model.*;  
  
public void getBucketPolicy(String bucketArn) {  
  
    GetBucketPolicyRequest reqGetBucketPolicy = new GetBucketPolicyRequest()  
        .withAccountId(AccountId)  
        .withBucket(bucketArn);  
  
    GetBucketPolicyResult respGetBucketPolicy =  
s3ControlClient.getBucketPolicy(reqGetBucketPolicy);  
    System.out.printf("GetBucketPolicy Response: %s%n",  
respGetBucketPolicy.toString());  
  
}
```

Löschen der Bucket-Richtlinie für Ihren Amazon-S3-on-Outposts-Bucket

Eine Bucket-Richtlinie ist eine auf Ressourcen basierende AWS Identity and Access Management (IAM)-Richtlinie, die Sie verwenden können, um Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte zu erteilen. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt. Weitere Informationen finden Sie unter [Bucket-Richtlinie](#).

In den folgenden Themen erfahren Sie, wie Sie Ihre Richtlinie für Amazon-S3-on-Outposts-Buckets mithilfe der AWS Management Console oder der AWS Command Line Interface (AWS CLI) anzeigen.

Verwenden der S3-Konsole

Löschen einer Bucket-Richtlinie

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, dessen Berechtigung Sie bearbeiten möchten.
4. Wählen Sie die Registerkarte Permissions (Berechtigungen) aus.
5. Wählen Sie im Bereich Outposts-Bucket-Richtlinie die Option Löschen aus.
6. Bestätigen Sie das Löschen.

Verwendung der AWS CLI

Im folgenden Beispiel wird die Bucket-Richtlinie für einen S3-on-Outposts-Bucket (s3-outposts:DeleteBucket) mithilfe der AWS CLI gelöscht. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

Beispiele für Bucket-Richtlinien

Mit S3-on-Outposts-Bucket-Richtlinien können Sie den Zugriff auf Objekte in Ihren S3-on-Outposts-Buckets sichern, sodass nur Benutzer mit den entsprechenden Berechtigungen darauf zugreifen können. Sie können sogar verhindern, dass authentifizierte Benutzer ohne die entsprechenden Berechtigungen auf Ihre S3-on-Outposts-Ressourcen zugreifen.

Dieser Abschnitt enthält Beispiele für typische Anwendungsfälle für S3-on-Outposts-Bucket-Richtlinien. Wenn Sie diese Richtlinien testen möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen (z. B. Ihren Bucket-Namen).

Wenn Sie einer Gruppe von Objekten Berechtigungen erteilen oder verweigern möchten, können Sie Platzhalterzeichen für (*) Amazon-Ressourcennamen (ARNs) und andere Werte verwenden. Sie können beispielsweise den Zugriff auf Gruppen von Objekten steuern, die mit einem gemeinsamen [Präfix](#) beginnen oder mit einer bestimmten Erweiterung wie `.html` enden.

Weitere Informationen zur AWS Identity and Access Management (IAM)-Richtliniensprache finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

Note

Wenn Sie [s3outposts](#) Berechtigungen mithilfe der Amazon S3-Konsole testen, müssen Sie zusätzliche Berechtigungen gewähren, die die Konsole benötigt, z. B. `s3outposts:createendpoint`, `s3outposts:listendpoints` usw.

Zusätzliche Ressourcen für die Erstellung von Bucket-Richtlinien

- Eine Liste der IAM-Richtlinienaktionen, Ressourcen und Bedingungsschlüssel, die Sie beim Erstellen einer S3-on-Outposts-Bucket-Richtlinie verwenden können, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 on Outposts](#).
- Hinweise zum Erstellen Ihrer S3-on-Outposts-Richtlinie finden Sie unter [Hinzufügen oder Bearbeiten einer Bucket-Richtlinie für einen Amazon-S3-on-Outposts-Bucket](#).

Themen

- [Verwalten des Zugriffs auf einen Amazon S3-on-Outposts-Bucket basierend auf bestimmten IP-Adressen](#)

Verwalten des Zugriffs auf einen Amazon S3-on-Outposts-Bucket basierend auf bestimmten IP-Adressen

Eine Bucket-Richtlinie ist eine auf Ressourcen basierende AWS Identity and Access Management (IAM)-Richtlinie, die Sie verwenden können, um Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte zu erteilen. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt. Weitere Informationen finden Sie unter [Bucket-Richtlinie](#).

Beschränken des Zugriffs auf bestimmte IP-Adressen

Das folgende Beispiel verweigert allen Benutzern das Ausführen von [S3-on-Outposts-Operationen](#) für Objekte in den angegebenen Buckets, es sei denn, die Anforderung stammt aus dem angegebenen IP-Adressbereich.

 Note

Wenn Sie den Zugriff auf eine bestimmte IP-Adresse einschränken, stellen Sie sicher, dass Sie auch angeben, welche VPC-Endpunkte, VPC-Quell-IP-Adressen oder externen IP-Adressen auf den S3-on-Outposts-Bucket zugreifen können. Andernfalls verlieren Sie möglicherweise den Zugriff auf den Bucket, wenn Ihre Richtlinie allen Benutzern die Ausführung von [s3outposts](#) Vorgängen an Objekten in Ihrem S3-on-Outposts-Bucket verweigert, ohne dass bereits die entsprechenden Berechtigungen vorhanden sind.

Die Condition-Anweisung dieser Richtlinie identifiziert **192.0.2.0/24** als Bereich zulässiger IP-Adressen der Version 4 (IPv4).

Der Condition-Block verwendet die Bedingungen `NotIpAddress` und den Bedingungsschlüssel `aws:SourceIp`, wobei es sich um einen AWS-übergreifenden Bedingungsschlüssel handelt. Der `aws:SourceIp`-Bedingungsschlüssel kann nur für öffentliche IP-Adressbereiche verwendet werden. Weitere Informationen zu diesen Bedingungsschlüsseln finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für S3 on Outposts](#). Die `aws:SourceIp-IPv4`-Werte verwenden die CIDR-Standardnotation. Weitere Informationen finden Sie in der [Referenz zu IAM-JSON-Richtlinienelementen](#) im IAM-Benutzerhandbuch.

 Warning

Bevor Sie diese S3-on-Outposts-Richtlinie verwenden, ersetzen Sie den **192.0.2.0/24** IP-Adressbereich in diesem Beispiel durch einen geeigneten Wert für Ihren Anwendungsfall. Andernfalls verlieren Sie die Möglichkeit, auf Ihren Bucket zuzugreifen.

```
{
  "Version": "2012-10-17",
  "Id": "S3OutpostsPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3outposts:*",
      "Resource": [
```

```

        "arn:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
accesspoint/EXAMPLE-ACCESS-POINT-NAME"
        "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET"
    ],
    "Condition": {
        "NotIpAddress": {
            "aws:SourceIp": "192.0.2.0/24"
        }
    }
}
]
}

```

Zulassen von IPv4- und IPv6-Adressen

Wenn Sie mit der Verwendung von IPv6-Adressen beginnen, empfehlen wir, dass Sie alle Richtlinien Ihrer Organisation zusätzlich zu Ihren bereits vorhandenen IPv4-Adressbereichen auf Ihre IPv6-Adressbereiche aktualisieren. Auf diese Weise können Sie sicherstellen, dass die Richtlinien während der Umstellung auf IPv6 weiterhin funktionieren.

Die folgende Bucket-Beispielrichtlinie für S3 on Outposts zeigt, wie Sie IPv4- und IPv6-Adressbereiche kombinieren, um alle gültigen IP-Adressen Ihrer Organisation abzudecken. Die Beispielrichtlinie erteilt Zugriff auf die IP-Adressen `192.0.2.1` und `2001:DB8:1234:5678::1` und verweigert den Zugriff auf die Adressen `203.0.113.1` und `2001:DB8:1234:5678:ABCD::1`.

Der `aws:SourceIp`-Bedingungs Schlüssel kann nur für öffentliche IP-Adressbereiche verwendet werden. Die IPv6-Werte für `aws:SourceIp` müssen im CIDR-Standardformat angegeben werden. Für IPv6 unterstützen wir die Verwendung von `::` zur Darstellung eines Bereichs von Nullen (z. B. `2001:DB8:1234:5678::/64`). Weitere Informationen finden Sie unter [IP-Adressen-Bedingungsoperatoren](#) im IAM-Benutzerhandbuch.

Warning

Ersetzen Sie die IP-Adressbereiche in diesem Beispiel durch die entsprechenden Werte für Ihren Anwendungsfall, bevor Sie diese S3-on-Outposts-Richtlinie verwenden. Andernfalls verlieren Sie möglicherweise die Möglichkeit, auf Ihren Bucket zuzugreifen.

```
{
```

```

    "Id": "S3OutpostsPolicyId2",
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "AllowIPmix",
        "Effect": "Allow",
        "Principal": "*",
        "Action": "s3outposts:*",
        "Resource": [
          "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET",
          "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-
ID/bucket/DOC-EXAMPLE-BUCKET/*"
        ],
        "Condition": {
          "IpAddress": {
            "aws:SourceIp": [
              "192.0.2.0/24",
              "2001:DB8:1234:5678::/64"
            ]
          },
          "NotIpAddress": {
            "aws:SourceIp": [
              "203.0.113.0/24",
              "2001:DB8:1234:5678:ABCD::/80"
            ]
          }
        }
      }
    ]
  }

```

Auflisten von Amazon-S3-on-Outposts-Buckets

Mit Amazon S3 in Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive

Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Weitere Informationen über das Arbeiten mit Buckets in S3 on Outposts finden Sie unter [Arbeiten mit S3-on-Outposts-Buckets](#).

Die folgenden Beispiele veranschaulichen, wie Sie eine Liste Ihrer S3-on-Outposts-Buckets mit der AWS Management Console, der AWS CLI und AWS SDK for Java zurückgeben.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Sehen Sie sich Ihre Liste der S3-on-Outposts-Buckets unter Outposts buckets (Outposts-Buckets) an.

Verwendung der AWS CLI

Im folgenden AWS CLI-Beispiel wird eine Liste von Buckets in einem Outpost abgerufen. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen zu diesem Befehl finden Sie unter [list-regional-buckets](#) in der AWS CLI-Referenz.

```
aws s3control list-regional-buckets --account-id 123456789012 --outpost-id op-01ac5d28a6a232904
```

Verwenden des AWS-SDKs für Java

Im folgenden SDK für Java-Beispiel wird eine Liste von Buckets in einem Outpost abgerufen. Weitere Informationen finden Sie unter [ListRegionalBuckets](#) in der API-Referenz zu Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;

public void listRegionalBuckets() {

    ListRegionalBucketsRequest reqListBuckets = new ListRegionalBucketsRequest()
```

```
        .withAccountId(AccountId)
        .withOutpostId(OutpostId);

    ListRegionalBucketsResult respListBuckets =
s3ControlClient.listRegionalBuckets(reqListBuckets);
    System.out.printf("ListRegionalBuckets Response: %s%n",
respListBuckets.toString());
}
```

Abrufen eines S3-on-Outposts-Buckets mithilfe der AWS CLI und des SDK for Java

Mit Amazon S3 in Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Die folgenden Beispiele veranschaulichen, wie Sie mithilfe der AWS CLI und AWS SDK for Java einen S3-on-Outposts-Bucket abrufen.

Note

Wenn Sie mit Amazon S3 on Outposts über die AWS CLI oder AWS-SDKs arbeiten, geben Sie den Zugriffspunkt-ARN für den Outpost anstelle des Bucket-Namens an. Der Zugriffspunkt-ARN nimmt das folgende Format an, wobei *region* der AWS-Region-Code für die Region ist, in der sich der Outpost befindet:

```
arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
accesspoint/example-outposts-access-point
```

Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Verwendung der AWS CLI

Im folgenden Beispiel wird ein S3-on-Outposts-Bucket mit der AWS CLI abgerufen. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [get-bucket](#) in der AWS CLI-Referenz.

```
aws s3control get-bucket --account-id 123456789012 --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket"
```

Verwenden des AWS-SDKs für Java

Im folgenden Beispiel für S3 on Outposts wird ein Bucket mit dem SDK for Java abgerufen. Weitere Informationen finden Sie unter [GetBucket](#) in der API-Referenz zu Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;

public void getBucket(String bucketArn) {

    GetBucketRequest reqGetBucket = new GetBucketRequest()
        .withBucket(bucketArn)
        .withAccountId(AccountId);

    GetBucketResult respGetBucket = s3ControlClient.getBucket(reqGetBucket);
    System.out.printf("GetBucket Response: %s%n", respGetBucket.toString());
}
```

Löschen Ihres Amazon-S3-on-Outposts-Buckets

Mit Amazon S3 in Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS

Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Weitere Informationen über das Arbeiten mit Buckets in S3 on Outposts finden Sie unter [Arbeiten mit S3-on-Outposts-Buckets](#).

Das AWS-Konto, das den Bucket erstellt, besitzt ihn und ist das einzige, das ihn löschen kann.

Note

- Outposts-Buckets müssen leer sein, bevor sie gelöscht werden können.

Die Amazon-S3-Konsole unterstützt keine S3-on-Outposts-Objektaktionen. Wenn Sie Objekte in Ihren S3-on-Outposts-Bucket hochladen und verwalten möchten, können Sie die REST-API, die AWS CLI oder AWS-SDKs verwenden.

- Bevor Sie einen Outposts-Bucket löschen können, müssen Sie alle Outposts-Zugriffspunkte für den Bucket löschen. Weitere Informationen finden Sie unter [Löschen eines Zugriffspunkts](#).
- Sie können einen Bucket nicht wiederherstellen, nachdem er gelöscht wurde.

Die folgenden Beispiele veranschaulichen, wie Sie einen S3-on-Outposts-Bucket mithilfe der AWS Management Console und der AWS Command Line Interface (AWS CLI) löschen.

Verwenden der S3-Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Bucket, den Sie löschen möchten, und wählen Sie Delete (Löschen).
4. Bestätigen Sie das Löschen.

Verwendung der AWS CLI

Im folgenden Beispiel wird ein S3-on-Outposts-Bucket (`s3-outposts:DeleteBucket`) mithilfe der AWS CLI gelöscht. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-bucket --account-id 123456789012 --bucket arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-  
bucket
```

Arbeiten mit Zugriffspunkten von Amazon S3 on Outposts

Um auf Ihren Amazon-S3-on-Outposts-Bucket zuzugreifen, müssen Sie einen Zugriffspunkt erstellen und konfigurieren.

Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie Amazon S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Zugriffspunkte unterstützen ausschließlich die virtuelle, Host-artige Adressierung.

Note

Das AWS-Konto, das den Outposts-Bucket erstellt, besitzt ihn und ist das einzige, das ihm Zugriffspunkte zuweisen kann.

In den folgenden Abschnitten wird beschrieben, wie Sie die Zugriffspunkte für S3-on-Outposts-Buckets erstellen und verwalten.

Themen

- [Erstellen eines S3-on-Outposts-Zugriffspunkten](#)
- [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#)
- [Anzeigen von Informationen über eine Zugriffspunktkonfiguration](#)
- [Eine Liste Ihrer Amazon-S3-on-Outposts-Zugriffspunkte anzeigen](#)
- [Löschen eines Zugriffspunkts](#)
- [Hinzufügen oder Bearbeiten einer Zugriffspunktrichtlinie](#)
- [Anzeigen einer Zugriffspunktrichtlinie für einen S3-on-Outposts-Zugriffspunkt](#)

Erstellen eines S3-on-Outposts-Zugriffspunkten

Um auf Ihren Amazon-S3-on-Outposts-Bucket zuzugreifen, müssen Sie einen Zugriffspunkt erstellen und konfigurieren.

Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie Amazon S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Zugriffspunkte unterstützen ausschließlich die virtuelle, Host-artige Adressierung.

In den folgenden Beispielen wird das Erstellen eines Zugriffspunkts für S3 on Outposts mithilfe der AWS Management Console, der AWS Command Line Interface (AWS CLI) und AWS SDK for Java veranschaulicht.

Note

Das AWS-Konto, das den Outposts-Bucket erstellt, besitzt ihn und ist das einzige, das ihm Zugriffspunkte zuweisen kann.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, für den Sie einen Outposts-Zugriffspunkt erstellen möchten.
4. Wählen Sie die Registerkarte Outposts-Zugriffspunkte.
5. Wählen Sie im Abschnitt Outposts-Zugriffspunkte die Option Outposts-Zugriffspunkte erstellen aus.
6. Geben Sie im Abschnitt Outposts access point settings (Einstellungen für den Outposts-Zugriffspunkt) einen Namen für den Zugriffspunkt ein und wählen Sie die Virtual Private Cloud (VPC) für den Zugriffspunkt aus.
7. Wenn Sie eine Richtlinie für Ihren Zugriffspunkt hinzufügen möchten, geben Sie sie in den Abschnitt Richtlinien für den Outposts-Zugriffspunkt ein.

Weitere Informationen finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

Verwendung der AWS CLI

Example

Im folgenden AWS CLI-Beispiel wird ein Zugriffspunkt für einen Outposts-Bucket erstellt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-access-point --account-id 123456789012
  --name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

Verwenden des AWS-SDKs für Java

Example

Im folgenden Beispiel für SDK für Java wird ein Zugriffspunkt für einen Outposts-Bucket erstellt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.services.s3control.model.*;

public String createAccessPoint(String bucketArn, String accessPointName) {

    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withName(accessPointName)
        .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP =
s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s\n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();
}
```

Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets

Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Jedes Mal, wenn Sie einen Zugriffspunkt für einen Bucket erstellen, generiert S3 on Outposts automatisch einen Zugriffspunkt-Alias. Sie können diesen Zugriffspunkt-Alias anstelle eines Zugriffspunkt-ARNs für jede Datenebenen-Operation verwenden. Sie können beispielsweise einen Zugriffspunkt-Alias verwenden, um Operationen auf Objektebene wie PUT, GET, LIST und mehr auszuführen. Eine Liste dieser Vorgänge finden Sie unter [Amazon-S3-API-Vorgänge für die Objektverwaltung](#).

Das folgende Beispiel zeigt einen ARN- und Zugriffspunkt-Alias für einen Zugriffspunkt namens *my-access-point*.

- Zugriffspunkt-ARN – `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/my-access-point`
- Zugriffspunkt-Alias – `my-access-po-o01ac5d28a6a232904e8xz5w8ijx1qz1bp3i3kuse10--op-s3`

Weitere Informationen zur Verwendung von ARNs finden Sie unter [Amazon-Ressourcennamen \(ARN\)](#) im Allgemeine AWS-Referenz.

Weitere Informationen über die Zugriffspunkt-Aliasse finden Sie in den folgenden Themen.

Themen

- [Zugriffspunkt-Aliasse](#)
- [Verwenden eines Zugriffspunkt-Alias in einer Objektoperation von S3 on Outposts](#)
- [Einschränkungen](#)

Zugriffspunkt-Aliasse

Ein Zugriffspunkt-Alias wird innerhalb desselben Namespace wie ein S3-on-Outposts-Bucket erstellt. Wenn Sie einen Zugriffspunkt erstellen, generiert S3 on Outposts automatisch einen Zugriffspunkt-Alias, der nicht geändert werden kann. Ein Zugriffspunkt-Alias erfüllt alle Anforderungen eines gültigen Bucket-Namens von S3 on Outposts und besteht aus den folgenden Teilen:

access point name prefix-metadata--op-s3

Note

Das Suffix `--op-s3` ist für Zugriffspunkt-Alias reserviert. Wir empfehlen, es nicht für Bucket- oder Zugriffspunktnamen zu verwenden. Weitere Informationen zu Bucket-Benennungsregeln für S3 on Outposts finden Sie unter [Arbeiten mit S3-on-Outposts-Buckets](#).

Suchen des Zugriffspunkt-Alias

Die folgenden Beispiele zeigen Ihnen, wie Sie einen Zugriffspunkt-Alias mit der Amazon-S3-Konsole und der AWS CLI finden.

Example : Suchen und Kopieren eines Zugriffspunkt-Alias in der Amazon-S3-Konsole

Nachdem Sie einen Zugriffspunkt in der Konsole erstellt haben, können Sie den Zugriffspunkt-Alias der Spalte Access Point alias (Zugriffspunkt-Alias) der Liste Access Points (Zugriffspunkte) entnehmen.

So kopieren Sie einen Zugriffspunkt-Alias

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts access points (Outposts--Zugriffspunkte) aus.
3. Zum Kopieren des Zugriffspunkt-Alias führen Sie einen der folgenden Schritte aus:
 - Wählen Sie in der Liste Access Points (Zugriffspunkte) das Optionsfeld neben dem Namen des Zugriffspunkts und dann Copy Access Point alias (Zugriffspunkt-Alias kopieren) aus.
 - Wählen Sie den Namen des Zugriffspunkts aus. Kopieren Sie dann unter Outposts access point overview (Outposts-Zugriffspunkt – Übersicht) den Zugriffspunkt-Alias.

Example : Erstellen eines Zugriffspunkts mit der AWS CLI und Suchen des Zugriffspunkt-Alias in der Antwort

Im folgenden AWS CLI-Beispiel für den `create-access-point`-Befehl wird der Zugriffspunkt erstellt und der automatisch generierte Zugriffspunkt-Alias zurückgegeben. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-access-point --bucket example-outposts-bucket --name example-outposts-access-point --account-id 123456789012
```

```
{
  "AccessPointArn":
    "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
accesspoint/example-outposts-access-point",
  "Alias": "example-outp-o01ac5d28a6a232904e8xz5w8ijx1qzlb3i3kuse10--op-s3"
}
```

Example : Abrufen eines Zugriffspunkt-Alias mithilfe der AWS CLI

Das folgende AWS CLI-Beispiel für den `get-access-point`-Befehl gibt Informationen über den angegebenen Zugriffspunkt zurück. Diese Informationen enthalten den Zugriffspunkt-Alias. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-access-point --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --name example-outposts-access-point --account-id 123456789012
```

```
{
  "Name": "example-outposts-access-point",
  "Bucket": "example-outposts-bucket",
  "NetworkOrigin": "Vpc",
  "VpcConfiguration": {
    "VpcId": "vpc-01234567890abcdef"
  },
  "PublicAccessBlockConfiguration": {
    "BlockPublicAcls": true,
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true
  },
  "CreationDate": "2022-09-18T17:49:15.584000+00:00",
  "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlb3i3kuse10--op-s3"
}
```

Example : Auflisten der Zugriffspunkte, um einen Zugriffspunkt-Alias mithilfe der AWS CLI zu finden

Das folgende AWS CLI-Beispiel für den `list-access-points`-Befehl listet Informationen über den angegebenen Zugriffspunkt auf. Diese Informationen enthalten den Zugriffspunkt-Alias. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.


```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket

{
  "AccessPointList": [
    {
      "Name": "example-outposts-access-point",
      "NetworkOrigin": "Vpc",
      "VpcConfiguration": {
        "VpcId": "vpc-01234567890abcdef"
      },
      "Bucket": "example-outposts-bucket",
      "AccessPointArn": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point",
      "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlp3i3kuse10--op-s3"
    }
  ]
}
```

Verwenden eines Zugriffspunkt-Alias in einer Objektoperation von S3 on Outposts

Bei der Übernahme von Zugriffspunkten können Sie Zugriffspunkt-Aliasse verwenden, ohne dass umfangreiche Codeänderungen erforderlich sind.

Dieses AWS CLI-Beispiel zeigt eine `get-object`-Operation für einen Bucket von S3 on Outposts. In diesem Beispiel wird anstelle des vollständigen Zugriffspunkt-ARN der Zugriffspunkt-Alias als Wert für `--bucket` verwendet.

```
aws s3api get-object --bucket my-access-po-
o0b1d075431d83bebde8xz5w8ijx1qzlp3i3kuse10--op-s3 --key testkey sample-object.rtf

{
  "AcceptRanges": "bytes",
  "LastModified": "2020-01-08T22:16:28+00:00",
  "ContentLength": 910,
  "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
  "VersionId": "null",
  "ContentType": "text/rtf",
  "Metadata": {}
}
```

Einschränkungen

- Aliase können nicht von Kunden konfiguriert werden.
- Aliasse können auf einem Zugriffspunkt nicht gelöscht, geändert oder deaktiviert werden.
- Sie können einen Zugriffspunkt-Alias nicht für Kontrollebenen-Operationen von S3 on Outposts verwenden. Eine Liste von Steuerebenen-Operationen von S3 on Outposts finden Sie unter [Amazon-S3-Control-API-Vorgänge zum Verwalten von Buckets](#).
- Aliasse können in AWS Identity and Access Management (IAM)-Richtlinien nicht verwendet werden.

Anzeigen von Informationen über eine Zugriffspunktconfiguration

Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie Amazon S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Zugriffspunkte unterstützen ausschließlich die virtuelle, Host-artige Adressierung.

In den folgenden Themen erfahren Sie, wie Sie Konfigurationsinformationen für einen S3-on-Outposts-Zugriffspunkt mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) und AWS SDK for Java zurückgeben können.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts access points (Outposts--Zugriffspunkte) aus.
3. Wählen Sie den Outposts-Zugriffspunkt aus, für den Sie Konfigurationsdetails anzeigen möchten.
4. Sehen Sie sich unter Outposts access point overview (Übersicht über Outposts-Zugriffspunkte) die Konfigurationsdetails zum Zugriffspunkt an.

Verwendung der AWS CLI

Das folgende AWS CLI-Beispiel ruft einen Zugriffspunkt für einen Outposts-Bucket ab. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-access-point --account-id 123456789012 --name arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-  
access-point
```

Verwenden des AWS-SDKs für Java

Im folgenden Beispiel für SDK für Java wird ein Zugriffspunkt für einen Outposts-Bucket abgerufen.

```
import com.amazonaws.services.s3control.model.*;  
  
public void getAccessPoint(String accessPointArn) {  
  
    GetAccessPointRequest reqGetAP = new GetAccessPointRequest()  
        .withAccountId(AccountId)  
        .withName(accessPointArn);  
  
    GetAccessPointResult respGetAP = s3ControlClient.getAccessPoint(reqGetAP);  
    System.out.printf("GetAccessPoint Response: %s%n", respGetAP.toString());  
  
}
```

Eine Liste Ihrer Amazon-S3-on-Outposts-Zugriffspunkte anzeigen

Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie Amazon S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Zugriffspunkte unterstützen ausschließlich die virtuelle, Host-artige Adressierung.

Die folgenden Themen zeigen Ihnen, wie Sie eine Liste Ihrer S3-on-Outposts-Zugriffspunkte mit der AWS Management Console, AWS Command Line Interface (AWS CLI) und AWS SDK for Java auflisten.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts access points (Outposts--Zugriffspunkte) aus.
3. Sehen Sie sich Ihre Liste der S3-on-Outposts-Zugriffspunkte unter Outposts access points (Outposts-Zugriffspunkte) an.

Verwendung der AWS CLI

Das folgende AWS CLI-Beispiel listet Zugriffspunkte für einen Outposts-Bucket auf. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

Verwenden des AWS-SDKs für Java

Im folgenden Beispiel für SDK für Java werden Zugriffspunkte für einen Outposts-Bucket aufgelistet.

```
import com.amazonaws.services.s3control.model.*;

public void listAccessPoints(String bucketArn) {

    ListAccessPointsRequest reqListAPs = new ListAccessPointsRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn);

    ListAccessPointsResult respListAPs = s3ControlClient.listAccessPoints(reqListAPs);
    System.out.printf("ListAccessPoints Response: %s\n", respListAPs.toString());

}
```

Löschen eines Zugriffspunkts

Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie Amazon S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Zugriffspunkte unterstützen ausschließlich die virtuelle, Host-artige Adressierung.

Die folgenden Beispiele zeigen Ihnen, wie Sie einen Zugriffspunkt mit der AWS Management Console und der AWS Command Line Interface (AWS CLI) löschen.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie im Navigationsbereich Outposts access points (Outposts-Zugriffspunkte) aus.
3. Wählen Sie im Bereich Outposts-Zugriffspunkte den Outposts-Zugriffspunkt aus, den Sie löschen möchten.
4. Wählen Sie Delete (Löschen).
5. Bestätigen Sie das Löschen.

Verwendung der AWS CLI

Im folgenden AWS CLI-Beispiel wird ein Outposts-Zugriffspunkt gelöscht. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-access-point --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Hinzufügen oder Bearbeiten einer Zugriffspunktrichtlinie

Jeder Zugriffspunkt verfügt über unterschiedliche Berechtigungen und Netzwerkkontrollen, die Amazon S3 on Outposts auf alle Anforderungen anwendet, die über diesen Zugriffspunkt eingehen. Jeder Zugriffspunkt erzwingt eine benutzerdefinierte Zugriffspunktrichtlinie, die in Verbindung mit der Bucket-Richtlinie funktioniert, die dem zugrunde liegenden Bucket zugeordnet ist. Weitere Informationen finden Sie unter [Zugriffspunkte](#).

Die folgenden Themen zeigen Ihnen, wie Sie die Zugriffspunktrichtlinie für S3 on Outposts mithilfe von AWS Management Console, AWS Command Line Interface (AWS CLI) und AWS SDK for Java hinzufügen oder bearbeiten.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, für den Sie die Zugriffspunktrichtlinie bearbeiten möchten.
4. Wählen Sie die Registerkarte Outposts-Zugriffspunkte.
5. Wählen Sie im Abschnitt Outposts-Zugriffspunkte den Zugriffspunkt aus, dessen Richtlinie Sie bearbeiten möchten, und wählen Sie Richtlinie bearbeiten.

6. Fügen Sie die Richtlinie im Abschnitt Richtlinien für den Outposts-Zugriffspunkt hinzu oder bearbeiten Sie sie. Weitere Informationen finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

Verwendung der AWS CLI

Im folgenden AWS CLI-Beispiel wird eine Richtlinie für einen Outposts-Zugriffspunkt eingerichtet.

1. Speichern Sie die folgende Zugriffspunktrichtlinie in einer JSON-Datei. In diesem Beispiel heißt die Datei `appolicy1.json`. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
{
  "Version": "2012-10-17",
  "Id": "exampleAccessPointPolicy",
  "Statement": [
    {
      "Sid": "st1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point"
    }
  ]
}
```

2. Senden Sie die JSON-Datei als Teil des CLI-Befehls `put-access-point-policy`. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control put-access-point-policy --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --policy file://appolicy1.json
```

Verwenden des AWS-SDKs für Java

Im folgenden SDK-für-Java-Beispiel wird eine Richtlinie für einen Outposts-Zugriffspunkt eingerichtet.

```
import com.amazonaws.services.s3control.model.*;

public void putAccessPointPolicy(String accessPointArn) {

    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testAccessPointPolicy\",
\"Statement\": [{\"Sid\":\"st1\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"" +
    AccountId + "\"},\"Action\":\"s3-outposts:*\",\"Resource\":\"" + accessPointArn +
    "\"}]}";

    PutAccessPointPolicyRequest reqPutAccessPointPolicy = new
    PutAccessPointPolicyRequest()
        .withAccountId(AccountId)
        .withName(accessPointArn)
        .withPolicy(policy);

    PutAccessPointPolicyResult respPutAccessPointPolicy =
    s3ControlClient.putAccessPointPolicy(reqPutAccessPointPolicy);
    System.out.printf("PutAccessPointPolicy Response: %s%n",
    respPutAccessPointPolicy.toString());
    printWriter.printf("PutAccessPointPolicy Response: %s%n",
    respPutAccessPointPolicy.toString());
}
```

Anzeigen einer Zugriffspunktrichtlinie für einen S3-on-Outposts-Zugriffspunkt

Jeder Zugriffspunkt verfügt über unterschiedliche Berechtigungen und Netzwerkkontrollen, die Amazon S3 on Outposts auf alle Anforderungen anwendet, die über diesen Zugriffspunkt eingehen. Jeder Zugriffspunkt erzwingt eine benutzerdefinierte Zugriffspunktrichtlinie, die in Verbindung mit der Bucket-Richtlinie funktioniert, die dem zugrunde liegenden Bucket zugeordnet ist. Weitere Informationen finden Sie unter [Zugriffspunkte](#).

Weitere Informationen zum Arbeiten mit Zugriffspunkten in S3 on Outposts finden Sie unter [Arbeiten mit S3-on-Outposts-Buckets](#).

Die folgenden Themen zeigen Ihnen, wie Sie die Zugriffspunktrichtlinie für S3 on Outposts mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) und AWS SDK for Java anzeigen.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.

2. Wählen Sie im Navigationsbereich Outposts access points (Outposts--Zugriffspunkte) aus.
3. Wählen Sie den Outposts-Zugriffspunkt aus, für den Sie die Richtlinie anzeigen möchten.
4. Überprüfen Sie auf dem Tab Permissions (Berechtigungen) die Zugriffspunktrichtlinie für S3 on Outposts.
5. Weitere Informationen zum Bearbeiten der Zugriffspunktrichtlinie finden Sie unter [Hinzufügen oder Bearbeiten einer Zugriffspunktrichtlinie](#).

Verwendung der AWS CLI

Im folgenden AWS CLI-Beispiel wird eine Richtlinie für einen Outposts-Zugriffspunkt abgerufen. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-access-point-policy --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Verwenden des AWS-SDKs für Java

Im folgenden SDK-für-Java-Beispiel wird eine Richtlinie für einen Outposts-Zugriffspunkt abgerufen.

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPointPolicy(String accessPointArn) {

    GetAccessPointPolicyRequest reqGetAccessPointPolicy = new
    GetAccessPointPolicyRequest()
        .withAccountId(AccountId)
        .withName(accessPointArn);

    GetAccessPointPolicyResult respGetAccessPointPolicy =
    s3ControlClient.getAccessPointPolicy(reqGetAccessPointPolicy);
    System.out.printf("GetAccessPointPolicy Response: %s%n",
    respGetAccessPointPolicy.toString());
    PrintWriter.printWriter.printf("GetAccessPointPolicy Response: %s%n",
    respGetAccessPointPolicy.toString());
}
```


Arbeiten mit Amazon-S3-on-Outposts-Endpunkten

Wenn Sie Anforderungen an einen Zugriffspunkt für Amazon S3 on Outposts weiterleiten möchten, müssen Sie einen S3-on-Outposts-Endpunkt erstellen und konfigurieren. Zum Erstellen eines Endpunkts muss Ihr Service-Link eine aktive Verbindung mit Ihrer Outposts-Heimatregion aufweisen. Jeder Virtual Private Cloud (VPC) in Ihrem Outpost kann ein Endpunkt zugeordnet sein. Weitere Informationen zu Endpunktkontingenten finden Sie unter [Netzwerkanforderungen von S3 on Outposts](#). Sie müssen einen Endpunkt erstellen, um auf Ihre Outposts-Buckets zugreifen und Objekt-Operationen ausführen zu können. Weitere Informationen finden Sie unter [Endpunkte](#).

Nachdem Sie einen Endpunkt erstellt haben, finden Sie im Feld „Status“ weitere Informationen zum Status des Endpunkts. Wenn Ihre Outposts offline sind, wird CREATE_FAILED zurückgegeben. Sie können Ihre Service-Link-Verbindung überprüfen, den Endpunkt löschen und das Erstellen erneut versuchen, nachdem die Verbindung wiederhergestellt wurde. Eine Liste mit zusätzlichen Fehlercodes finden Sie nachstehend. Weitere Informationen finden Sie unter [Endpunkte](#).

API	Status	Grund für Fehlschlag – Fehlercode	Meldung – Grund für Fehlschlag
CreateEndpoint	Create_Failed	OutpostNotReachable	Der Endpunkt konnte nicht erstellt werden, da die Service-Link-Verbindung zur Outposts-Heimatregion unterbrochen ist. Überprüfen Sie Ihre Verbindung, löschen Sie den Endpunkt und versuchen Sie es erneut.
CreateEndpoint	Create_Failed	InternalError	Der Endpunkt konnte aufgrund eines internen Fehlers nicht erstellt werden. Bitte löschen Sie den Endpunkt und erstellen Sie ihn erneut.
DeleteEndpoint	Delete_Failed	OutpostNotReachable	Der Endpunkt konnte nicht gelöscht werden, da die Service-Link-Verbindung zur Outposts-Heimatregion unterbrochen ist. Überprüfen Sie Ihre Verbindung und versuchen Sie es erneut.

API	Status	Grund für Fehlschlag – Fehlercode	Meldung – Grund für Fehlschlag
DeleteEndpoint	Delete_Failed	InternalError	Der Endpunkt konnte aufgrund eines internen Fehlers nicht gelöscht werden. Bitte versuchen Sie es noch einmal.

Weitere Informationen über das Arbeiten mit Buckets in S3 on Outposts finden Sie unter [Arbeiten mit S3-on-Outposts-Buckets](#).

In den folgenden Abschnitten wird die Erstellung und Verwaltung von Endpunkten für S3 on Outposts beschrieben.

Themen

- [Erstellen eines Endpunkts in einem Outpost](#)
- [Anzeigen einer Liste Ihrer Amazon-S3-on-Outposts-Endpunkte](#)
- [Löschen eines Amazon-S3-on-Outposts-Endpunkts](#)

Erstellen eines Endpunkts in einem Outpost

Wenn Sie Anforderungen an einen Zugriffspunkt für Amazon S3 on Outposts weiterleiten möchten, müssen Sie einen S3-on-Outposts-Endpunkt erstellen und konfigurieren. Zum Erstellen eines Endpunkts muss Ihr Service-Link eine aktive Verbindung mit Ihrer Outposts-Heimatregion aufweisen. Jeder Virtual Private Cloud (VPC) in Ihrem Outpost kann ein Endpunkt zugeordnet sein. Weitere Informationen zu Endpunktkontingenten finden Sie unter [Netzwerkanforderungen von S3 on Outposts](#). Sie müssen einen Endpunkt erstellen, um auf Ihre Outposts-Buckets zugreifen und Objekt-Operationen ausführen zu können. Weitere Informationen finden Sie unter [Endpunkte](#).

Berechtigungen

Weitere Informationen zu den erforderlichen Berechtigungen für das Erstellen eines Endpunkts finden Sie unter [Berechtigungen für S3-on-Outposts-Endpunkte](#).

Wenn Sie einen Endpunkt erstellen, erstellt S3 on Outposts auch eine serviceverknüpfte Rolle in Ihrem AWS-Konto. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für S3 on Outposts](#).

Die folgenden Beispiele zeigen, wie Sie einen S3-on-Outposts-Endpunkt mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) und AWS SDK for Java erstellen.

Verwenden der S3-Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts access points (Outposts-Zugriffspunkte) aus.
3. Wählen Sie den Tab Outposts endpoints (Outposts-Endpunkte) aus.
4. Wählen Sie Create Outposts endpoint (Outposts-Endpunkt erstellen) aus.
5. Wählen Sie unter Outpost den Outpost aus, auf dem dieser Endpunkt erstellt werden soll.
6. Wählen Sie unter VPC eine VPC aus, die noch keinen Endpunkt hat und außerdem den Regeln für Outposts-Endpunkte entspricht.

Eine Virtual Private Cloud (VPC) ermöglicht es Ihnen, AWS-Ressourcen in einem virtuellen Netzwerk zu launchen, das Sie definieren. Dieses virtuelle Netzwerk entspricht weitgehend einem herkömmlichen Netzwerk, wie Sie es in Ihrem Rechenzentrum betreiben würden, kann jedoch die Vorteile der skalierbaren Infrastruktur von AWS nutzen.

Wenn Sie keine VPC haben, wählen Sie VPC erstellen. Weitere Informationen finden Sie unter [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#).

7. Wählen Sie Create Outposts endpoint (Outposts-Endpunkt erstellen) aus.

Verwendung von AWS CLI

Example

Im folgenden AWS CLI-Beispiel wird ein Endpunkt für einen Outpost mithilfe des VPC-Ressourcenzugriffstyps erstellt. Die VPC ist vom Subnetz abgeleitet. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id  
subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

Im folgenden AWS CLI-Beispiel wird ein Endpunkt für einen Outpost mit dem Zugriffstyp des kundeneigenen IP-Adresspools (CoIP-Pool) erstellt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
  subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

Verwenden des AWS-SDKs für Java

Example

Im folgenden SDK für Java-Beispiel wird ein Endpunkt für einen Outpost erstellt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;

public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
        .withOutpostId("op-0d79779cef3c30a40")
        .withSubnetId("subnet-8c7a57c5")
        .withSecurityGroupId("sg-ab19e0d1")
        .withAccessType("CustomerOwnedIp")
        .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
    // Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type is
    // customer-owned IP address pool (CoIP pool)
    CreateEndpointResult createEndpointResult =
s3OutpostsClient.createEndpoint(createEndpointRequest);
    System.out.println("Endpoint is created and its ARN is " +
createEndpointResult.getEndpointArn());
}
```

Anzeigen einer Liste Ihrer Amazon-S3-on-Outposts-Endpunkte

Wenn Sie Anforderungen an einen Zugriffspunkt für Amazon S3 on Outposts weiterleiten möchten, müssen Sie einen S3-on-Outposts-Endpunkt erstellen und konfigurieren. Zum Erstellen eines Endpunkts muss Ihr Service-Link eine aktive Verbindung mit Ihrer Outposts-Heimatregion aufweisen. Jeder Virtual Private Cloud (VPC) in Ihrem Outpost kann ein Endpunkt zugeordnet sein. Weitere Informationen zu Endpunktkontingenten finden Sie unter [Netzwerkanforderungen von S3 on](#)

[Outposts](#). Sie müssen einen Endpunkt erstellen, um auf Ihre Outposts-Buckets zugreifen und Objekt-Operationen ausführen zu können. Weitere Informationen finden Sie unter [Endpunkte](#).

Die folgenden Beispiele veranschaulichen, wie Sie eine Liste Ihrer S3-on-Outposts-Endpunkte mit der AWS Management Console, der AWS Command Line Interface (AWS CLI) und AWS SDK for Java zurückgeben.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts access points (Outposts-Zugriffspunkte) aus.
3. Wählen Sie auf der Seite Outposts access points (Outposts-Zugriffspunkte) den Tab Outposts endpoints (Outposts-Endpunkte) aus.
4. Unter Outposts endpoints (Outposts-Endpunkte) können Sie eine Liste Ihrer S3-on-Outposts-Endpunkte anzeigen.

Verwendung von AWS CLI

Das folgende AWS CLI-Beispiel listet die Endpunkte für die AWS Outposts-Ressourcen auf, die Ihrem Konto zugeordnet sind. Weitere Informationen über diesen Befehl finden Sie unter [list-endpoints](#) in der AWS CLI-Referenz.

```
aws s3outposts list-endpoints
```

Verwenden des AWS-SDKs für Java

Im folgenden SDK für Java-Beispiel werden Endpunkte für einen Outpost aufgelistet. Weitere Informationen finden Sie unter [ListEndpoints](#) in der API-Referenz zu Amazon Simple Storage Service.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.ListEndpointsRequest;
import com.amazonaws.services.s3outposts.model.ListEndpointsResult;

public void listEndpoints() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    ListEndpointsRequest listEndpointsRequest = new ListEndpointsRequest();
```

```
ListEndpointsResult listEndpointsResult =
s3OutpostsClient.listEndpoints(listEndpointsRequest);
System.out.println("List endpoints result is " + listEndpointsResult);
}
```

Löschen eines Amazon-S3-on-Outposts-Endpunkts

Wenn Sie Anforderungen an einen Zugriffspunkt für Amazon S3 on Outposts weiterleiten möchten, müssen Sie einen S3-on-Outposts-Endpunkt erstellen und konfigurieren. Zum Erstellen eines Endpunkts muss Ihr Service-Link eine aktive Verbindung mit Ihrer Outposts-Heimatregion aufweisen. Jeder Virtual Private Cloud (VPC) in Ihrem Outpost kann ein Endpunkt zugeordnet sein. Weitere Informationen zu Endpunktkontingenten finden Sie unter [Netzwerkanforderungen von S3 on Outposts](#). Sie müssen einen Endpunkt erstellen, um auf Ihre Outposts-Buckets zugreifen und Objekt-Operationen ausführen zu können. Weitere Informationen finden Sie unter [Endpunkte](#).

Die folgenden Beispiele veranschaulichen, wie Sie Ihre S3-on-Outposts-Endpunkte mit der AWS Management Console, der AWS Command Line Interface (AWS CLI) und AWS SDK for Java löschen.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts access points (Outposts-Zugriffspunkte) aus.
3. Wählen Sie auf der Seite Outposts access points (Outposts-Zugriffspunkte) den Tab Outposts endpoints (Outposts-Endpunkte) aus.
4. Wählen Sie unter Outposts endpoints (Outposts-Endpunkte) den Endpunkt aus, den Sie löschen möchten, und klicken Sie dann auf Delete (Löschen).

Verwendung von AWS CLI

Im folgenden AWS CLI-Beispiel wird ein Endpunkt für einen Outpost gelöscht. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3outposts delete-endpoint --endpoint-id example-endpoint-id --outpost-
id op-01ac5d28a6a232904
```

Verwenden des AWS-SDKs für Java

Im folgenden SDK-für-Java-Beispiel wird ein Endpunkt für einen Outpost gelöscht. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.arn.Arn;
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.DeleteEndpointRequest;

public void deleteEndpoint(String endpointArnInput) {
    String outpostId = "op-01ac5d28a6a232904";
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    Arn endpointArn = Arn.fromString(endpointArnInput);
    String[] resourceParts = endpointArn.getResource().getResource().split("/");
    String endpointId = resourceParts[resourceParts.length - 1];
    DeleteEndpointRequest deleteEndpointRequest = new DeleteEndpointRequest()
        .withEndpointId(endpointId)
        .withOutpostId(outpostId);
    s3OutpostsClient.deleteEndpoint(deleteEndpointRequest);
    System.out.println("Endpoint with id " + endpointId + " is deleted.");
}
```

Arbeiten mit S3-on-Outposts-Objekten

Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihren AWS Outposts erstellen und Objekte für Anwendungen, die lokalen Datenzugriff, lokale Datenverarbeitung und Datenresidenz erfordern, einfach vor Ort speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse, S3 Outposts (OUTPOSTS), die Amazon S3 verwendet und darauf ausgelegt ist APIs, Daten dauerhaft und redundant auf mehreren Geräten und Servern auf Ihrem zu speichern. AWS Outposts Sie kommunizieren mit Ihrem Outpost-Bucket, indem Sie einen Access Point und eine Endpunktverbindung über eine virtuelle private Cloud (VPC) verwenden. Sie können für Outpost-Buckets dieselben APIs Funktionen wie für Amazon S3 S3-Buckets verwenden, einschließlich Zugriffsrichtlinien, Verschlüsselung und Tagging. Sie können S3 auf Outposts über das AWS Management Console, AWS Command Line Interface (AWS CLI) AWS SDKs, oder REST API verwenden.

Objekte sind die Grundeinheiten, die in Amazon S3 on Outposts gespeichert sind. Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Access Point Amazon Resource Name (ARN) oder den Access Point-Alias. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN Format für S3 auf Outposts-Zugriffspunkten, das den AWS-Region Code für die Region, in der der Outpost beheimatet ist, die ID, die AWS-Konto Outpost-ID und den Namen des Access Points enthält:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Weitere Informationen zu S3 on Outposts finden Sie ARNs unter [Ressource ARNs für S3 auf Outposts](#).

Das Objekt ARNs verwendet das folgende Format, das die Adresse, auf AWS-Region die sich der Outpost bezieht, die ID, die AWS-Konto Outpost-ID, den Bucket-Namen und den Objektschlüssel umfasst:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn Sie ein AWS Outpost-Rack installieren, bleiben Ihre Daten lokal in Ihrem Outpost, um die Anforderungen an die Datenresidenz zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da das in der Region gehostet AWS Management Console wird, können Sie die Konsole nicht zum Hochladen oder Verwalten von Objekten in Ihrem Outpost verwenden. Sie können jedoch das, AWS Command Line Interface (AWS CLI) und verwenden RESTAPI, AWS SDKs um Ihre Objekte über Ihre Access Points hochzuladen und zu verwalten.

Themen

- [Laden Sie ein Objekt in einen S3-Bucket auf Outposts hoch](#)
- [Kopieren eines Objekts in einem Amazon S3 on Outposts-Bucket mit AWS SDK for Java](#)
- [Abrufen eines Objekts aus einem Amazon-S3-on-Outposts-Bucket](#)
- [Auflisten der Objekten in einem Amazon-S3-on-Outposts-Bucket](#)
- [Löschen von Objekten in Amazon-S3-on-Outposts-Buckets](#)

- [Verwenden von HeadBucket, um festzustellen, ob ein S3-on-Outposts-Bucket vorhanden ist und Sie Zugriffsberechtigungen haben](#)
- [Durchführen und Verwalten eines mehrteiligen Uploads mit dem SDK for Java](#)
- [Verwenden vorsignierter URLs für S3 on Outposts](#)
- [Amazon S3 on Outposts mit lokalem Amazon EMR on Outposts](#)
- [Zwischenspeicherung von Autorisierung und Authentifizierung](#)

Laden Sie ein Objekt in einen S3-Bucket auf Outposts hoch

Objekte sind die Grundeinheiten, die in Amazon S3 on Outposts gespeichert sind. Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Amazon-Ressourcennamen (ARN) oder den Alias des Zugriffspunkts. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN-Format für S3 auf Outposts-Zugriffspunkten, das den AWS-Region Code für die Region, in der der Outpost beheimatet ist, die ID, die AWS-Konto Outpost-ID und den Namen des Access Points enthält:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn Sie ein AWS Outpost-Rack installieren, bleiben Ihre Daten lokal in Ihrem Outpost, um die Anforderungen an die Datenresidenz zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da das in der Region gehostet AWS Management Console wird, können Sie die Konsole nicht zum Hochladen oder Verwalten von Objekten in Ihrem Outpost verwenden. Sie können jedoch die REST-API AWS Command Line Interface (AWS CLI) und AWS SDKs verwenden, um Ihre Objekte über Ihre Access Points hochzuladen und zu verwalten.

Im Folgenden AWS CLI und in AWS SDK for Java Beispielen wird gezeigt, wie Sie mithilfe eines Access Points ein Objekt in einen S3 on Outposts-Bucket hochladen.

AWS CLI

Example

Im folgenden Beispiel wird ein Objekt mit dem Namen `sample-object.xml` in einen S3-on-Outposts-Bucket (`s3-outposts:PutObject`) mit der AWS CLI eingefügt. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [put-object](#) in der AWS CLI -Referenz.

```
aws s3api put-object --bucket arn:aws:s3-
outposts:Region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --key sample-object.xml --body sample-object.xml
```

SDK for Java

Example

Im folgenden Beispiel wird ein Objekt mithilfe des SDK for Java in einen S3-on-Outposts-Bucket eingefügt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen. Weitere Informationen finden Sie unter [Objekte hochladen](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;

public class PutObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
            credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
```

```
        .enableUseArnRegion()
        .build();

    // Upload a text string as a new object.
    s3Client.putObject(accessPointArn, stringObjKeyName, "Uploaded String
Object");

    // Upload a file as a new object with ContentType and title specified.
    PutObjectRequest request = new PutObjectRequest(accessPointArn,
fileObjKeyName, new File(fileName));
    ObjectMetadata metadata = new ObjectMetadata();
    metadata.setContentType("plain/text");
    metadata.addUserMetadata("title", "someTitle");
    request.setMetadata(metadata);
    s3Client.putObject(request);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

Kopieren eines Objekts in einem Amazon S3 on Outposts-Bucket mit AWS SDK for Java

Objekte sind die Grundeinheiten, die in Amazon S3 on Outposts gespeichert sind. Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Amazon-Ressourcennamen (ARN) oder den Alias des Zugriffspunkts. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN-Format für S3-on-Outposts-Objekte, das den AWS-Region-Code für die Region, in der sich der Outpost befindet, die AWS-Konto-ID, Outpost-ID, Bucket-Name und den Namen des Zugriffspunkts umfasst:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn AWS ein Outpost-Rack installiert, bleiben Ihre Daten in Ihrem Outpost lokal, um die Anforderungen an die Datenspeicherorte zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da die AWS Management Console innerhalb der Region gehostet wird, können Sie die Konsole nicht verwenden, um Objekte in Ihrem Outpost hochzuladen oder zu verwalten. Sie können die REST-API, AWS Command Line Interface (AWS CLI) und AWS-SDKs zum Hochladen und Verwalten Ihrer Objekte über Ihre Zugriffspunkte verwenden.

Das folgenden Beispiel veranschaulicht, wie Sie mithilfe von AWS SDK for Java ein Objekt in einem S3-on-Outposts-Bucket kopieren.

Verwenden des AWS-SDKs für Java

Im folgenden S3-on-Outposts-Beispiel wird ein Objekt mithilfe des SDK für Java in ein neues Objekt im selben Bucket kopiert. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

public class CopyObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String sourceKey = "*** Source object key ***";
        String destinationKey = "*** Destination object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();
```

```
        // Copy the object into a new object in the same bucket.
        CopyObjectRequest copyObjectRequest = new CopyObjectRequest(accessPointArn,
sourceKey, accessPointArn, destinationKey);
        s3Client.copyObject(copyObjectRequest);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

Abrufen eines Objekts aus einem Amazon-S3-on-Outposts-Bucket

Objekte sind die Grundeinheiten, die in Amazon S3 on Outposts gespeichert sind. Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Amazon-Ressourcennamen (ARN) oder den Alias des Zugriffspunkts. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN-Format für S3-on-Outposts-Objekte, das den AWS-Region-Code für die Region, in der sich der Outpost befindet, die AWS-Konto-ID, Outpost-ID, Bucket-Name und den Namen des Zugriffspunkts umfasst:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn AWS ein Outpost-Rack installiert, bleiben Ihre Daten in Ihrem Outpost lokal, um die Anforderungen an die Datenspeicherorte zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da die AWS Management Console innerhalb der Region gehostet wird, können Sie die Konsole nicht verwenden, um Objekte in Ihrem Outpost hochzuladen oder zu verwalten. Sie

können die REST-API, AWS Command Line Interface (AWS CLI) und AWS-SDKs zum Hochladen und Verwalten Ihrer Objekte über Ihre Zugriffspunkte verwenden.

Die folgenden Beispiele veranschaulichen, wie Sie ein Objekt mithilfe der AWS Command Line Interface (AWS CLI) und AWS SDK for Java herunterladen (abrufen).

Verwendung der AWS CLI

Im folgenden Beispiel wird ein Objekt mit dem Namen `sample-object.xml` in einem S3-on-Outposts-Bucket (`s3-outposts:GetObject`) mit der AWS CLI abgerufen. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [get-object](#) in der AWS CLI-Referenz.

```
aws s3api get-object --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --key testkey sample-object.xml
```

Verwenden des AWS-SDKs für Java

Im folgenden Beispiel für S3 on Outposts wird ein Objekt mit dem SDK for Java abgerufen. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen. Weitere Informationen finden Sie unter [GetObject](#) in der Amazon Simple Storage Service-API-Referenz.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ResponseHeaderOverrides;
import com.amazonaws.services.s3.model.S3Object;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class GetObject {
    public static void main(String[] args) throws IOException {
        String accessPointArn = "*** access point ARN ***";
        String key = "*** Object key ***";
```

```
S3Object fullObject = null, objectPortion = null, headerOverrideObject = null;
try {
    // This code expects that you have AWS credentials set up per:
    // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .enableUseArnRegion()
        .build();

    // Get an object and print its contents.
    System.out.println("Downloading an object");
    fullObject = s3Client.getObject(new GetObjectRequest(accessPointArn, key));
    System.out.println("Content-Type: " +
fullObject.getObjectMetadata().getContentType());
    System.out.println("Content: ");
    displayTextInputStream(fullObject.getObjectContent());

    // Get a range of bytes from an object and print the bytes.
    GetObjectRequest rangeObjectRequest = new GetObjectRequest(accessPointArn,
key)
        .withRange(0, 9);
    objectPortion = s3Client.getObject(rangeObjectRequest);
    System.out.println("Printing bytes retrieved.");
    displayTextInputStream(objectPortion.getObjectContent());

    // Get an entire object, overriding the specified response headers, and
print the object's content.
    ResponseHeaderOverrides headerOverrides = new ResponseHeaderOverrides()
        .withCacheControl("No-cache")
        .withContentDisposition("attachment; filename=example.txt");
    GetObjectRequest getObjectRequestHeaderOverride = new
GetObjectRequest(accessPointArn, key)
        .withResponseHeaders(headerOverrides);
    headerOverrideObject = s3Client.getObject(getObjectRequestHeaderOverride);
    displayTextInputStream(headerOverrideObject.getObjectContent());
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
} finally {
```

```
        // To ensure that the network connection doesn't remain open, close any
open input streams.
        if (fullObject != null) {
            fullObject.close();
        }
        if (objectPortion != null) {
            objectPortion.close();
        }
        if (headerOverrideObject != null) {
            headerOverrideObject.close();
        }
    }
}

private static void displayTextInputStream(InputStream input) throws IOException {
    // Read the text input stream one line at a time and display each line.
    BufferedReader reader = new BufferedReader(new InputStreamReader(input));
    String line = null;
    while ((line = reader.readLine()) != null) {
        System.out.println(line);
    }
    System.out.println();
}
}
```

Auflisten der Objekten in einem Amazon-S3-on-Outposts-Bucket

Objekte sind die Grundeinheiten, die in Amazon S3 on Outposts gespeichert sind. Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Amazon-Ressourcennamen (ARN) oder den Alias des Zugriffspunkts. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN-Format für S3-on-Outposts-Objekte, das den AWS-Region-Code für die Region, in der sich der Outpost befindet, die AWS-Konto-ID, Outpost-ID, Bucket-Name und den Namen des Zugriffspunkts umfasst:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```


Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Note

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn AWS ein Outpost-Rack installiert, bleiben Ihre Daten in Ihrem Outpost lokal, um die Anforderungen an die Datenspeicherorte zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da die AWS Management Console innerhalb der Region gehostet wird, können Sie die Konsole nicht verwenden, um Objekte in Ihrem Outpost hochzuladen oder zu verwalten. Sie können die REST-API, AWS Command Line Interface (AWS CLI) und AWS-SDKs zum Hochladen und Verwalten Ihrer Objekte über Ihre Zugriffspunkte verwenden.

Die folgenden Beispiele veranschaulichen, wie Sie mithilfe der AWS CLI und AWS SDK for Java die Objekte in einem S3-on-Outposts-Bucket auflisten.

Verwendung der AWS CLI

Im folgenden Beispiel werden die Objekte in einem S3-on-Outposts-Bucket (`s3-outposts:ListObjectsV2`) unter Verwendung der AWS CLI aufgelistet. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [list-objects-v2](#) in der AWS CLI-Referenz.

```
aws s3api list-objects-v2 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Note

Wenn Sie diese Aktion mit Amazon S3 auf Outposts über die AWS SDKs verwenden, geben Sie den Outposts-Zugangspunkt-ARN anstelle des Bucket-Namens in der folgenden Form an: `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-Outposts-Access-Point`. Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Verwenden des AWS-SDKs für Java

Das folgende S3-on-Outposts-Beispiel listet Objekte in einem Bucket mit dem SDK for Java auf. Zum Verwenden dieses Beispiels ersetzen Sie jeden *user input placeholder* durch Ihre eigenen Informationen.

Important

In diesem Beispiel verwenden wir [ListObjectsV2](#), welches die neueste Version der ListObjects-API-Operation ist. Wir empfehlen die Verwendung dieser überarbeiteten API-Operationen für die Anwendungsentwicklung. Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin die vorherige Version dieser API-Operation.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsV2Request;
import com.amazonaws.services.s3.model.ListObjectsV2Result;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class ListObjectsV2 {

    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            System.out.println("Listing objects");

            // maxKeys is set to 2 to demonstrate the use of
            // ListObjectsV2Result.getNextContinuationToken()
            ListObjectsV2Request req = new
ListObjectsV2Request().withBucketName(accessPointArn).withMaxKeys(2);
            ListObjectsV2Result result;
```

```
do {
    result = s3Client.listObjectsV2(req);

    for (S3ObjectSummary objectSummary : result.getObjectSummaries()) {
        System.out.printf(" - %s (size: %d)\n", objectSummary.getKey(),
objectSummary.getSize());
    }
    // If there are more than maxKeys keys in the bucket, get a
continuation token
    // and list the next objects.
    String token = result.getNextContinuationToken();
    System.out.println("Next Continuation Token: " + token);
    req.setContinuationToken(token);
} while (result.isTruncated());
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Löschen von Objekten in Amazon-S3-on-Outposts-Buckets

Objekte sind die Grundeinheiten, die in Amazon S3 on Outposts gespeichert sind. Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Amazon-Ressourcennamen (ARN) oder den Alias des Zugriffspunkts. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN-Format für S3-on-Outposts-Objekte, das den AWS-Region-Code für die Region, in der sich der Outpost befindet, die AWS-Konto-ID, Outpost-ID, Bucket-Name und den Namen des Zugriffspunkts umfasst:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn AWS ein Outpost-Rack installiert, bleiben Ihre Daten in Ihrem Outpost lokal, um die Anforderungen an die Datenspeicherorte zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da die AWS Management Console innerhalb der Region gehostet wird, können Sie die Konsole nicht verwenden, um Objekte in Ihrem Outpost hochzuladen oder zu verwalten. Sie können die REST-API, AWS Command Line Interface (AWS CLI) und AWS-SDKs zum Hochladen und Verwalten Ihrer Objekte über Ihre Zugriffspunkte verwenden.

Die folgenden Beispiele veranschaulichen, wie Sie ein einzelnes Objekt oder mehrere Objekte in einem S3-on-Outposts-Bucket mithilfe der AWS Command Line Interface (AWS CLI) und AWS SDK for Java löschen.

Verwendung der AWS CLI

Die folgenden Beispiele veranschaulichen, wie Sie ein einzelnes Objekt oder mehrere Objekte aus einem S3-on-Outposts-Bucket löschen.

delete-object

Im folgenden Beispiel wird ein Objekt mit dem Namen `sample-object.xml` in einem S3-on-Outposts-Bucket (`s3-outposts:DeleteObject`) mithilfe der AWS CLI gelöscht. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [delete-object](#) in der AWS CLI-Befehlsreferenz.

```
aws s3api delete-object --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --key sample-object.xml
```

delete-objects

Im folgenden Beispiel werden zwei Objekte mit dem Namen `sample-object.xml` und `test1.text` in einem S3-on-Outposts-Bucket (`s3-outposts:DeleteObject`) mithilfe der AWS CLI gelöscht. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen zu diesem Befehl finden Sie unter [delete-objects](#) in der AWS CLI-Referenz.

```
aws s3api delete-objects --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --delete file://delete.json
```

```
delete.json
{
  "Objects": [
    {
      "Key": "test1.txt"
    },
    {
      "Key": "sample-object.xml"
    }
  ],
  "Quiet": false
}
```

Verwenden des AWS-SDKs für Java

Die folgenden Beispiele veranschaulichen, wie Sie ein einzelnes Objekt oder mehrere Objekte aus einem S3-on-Outposts-Bucket löschen.

DeleteObject

Im folgenden Beispiel für S3 on Outposts wird ein Objekt in einem Bucket mit dem SDK for Java gelöscht. Zum Verwenden dieses Beispiels geben Sie den Zugriffspunkt-ARN für den Outpost und den Schlüsselnamen für das Objekt an, das Sie löschen möchten. Weitere Informationen finden Sie unter [DeleteObject](#) in der API-Referenz zum Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

public class DeleteObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** key name ****";

        try {
```

```
// This code expects that you have AWS credentials set up per:
// https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
    .enableUseArnRegion()
    .build();

s3Client.deleteObject(new DeleteObjectRequest(accessPointArn, keyName));
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

DeleteObjects

Das folgende S3-on-Outposts-Beispiel lädt Objekte in einem Bucket hoch und löscht sie dann mithilfe des SDK for Java. Wenn Sie dieses Beispiel verwenden möchten, geben Sie den Zugriffspunkt-ARN für den Outpost an. Weitere Informationen finden Sie unter [DeleteObjects](#) in der API-Referenz zu Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectsRequest;
import com.amazonaws.services.s3.model.DeleteObjectsRequest.KeyVersion;
import com.amazonaws.services.s3.model.DeleteObjectsResult;

import java.util.ArrayList;

public class DeleteObjects {

    public static void main(String[] args) {
        String accessPointArn = "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point";
```

```
try {
    // This code expects that you have AWS credentials set up per:
    // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .enableUseArnRegion()
        .build();

    // Upload three sample objects.
    ArrayList<KeyVersion> keys = new ArrayList<KeyVersion>();
    for (int i = 0; i < 3; i++) {
        String keyName = "delete object example " + i;
        s3Client.putObject(accessPointArn, keyName, "Object number " + i + "
to be deleted.");
        keys.add(new KeyVersion(keyName));
    }
    System.out.println(keys.size() + " objects successfully created.");

    // Delete the sample objects.
    DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest(accessPointArn)
        .withKeys(keys)
        .withQuiet(false);

    // Verify that the objects were deleted successfully.
    DeleteObjectsResult delObjRes =
s3Client.deleteObjects(multiObjectDeleteRequest);
    int successfulDeletes = delObjRes.getDeletedObjects().size();
    System.out.println(successfulDeletes + " objects successfully
deleted.");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Verwenden von HeadBucket, um festzustellen, ob ein S3-on-Outposts-Bucket vorhanden ist und Sie Zugriffsberechtigungen haben

Objekte sind die Grundeinheiten, die in Amazon S3 on Outposts gespeichert sind. Jedes Objekt ist in einem Bucket enthalten. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie den Bucket für Objektoperationen angeben, verwenden Sie den Amazon-Ressourcennamen (ARN) oder den Alias des Zugriffspunkts. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Das folgende Beispiel zeigt das ARN-Format für S3-on-Outposts-Objekte, das den AWS-Region-Code für die Region, in der sich der Outpost befindet, die AWS-Konto-ID, Outpost-ID, Bucket-Name und den Namen des Zugriffspunkts umfasst:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Weitere Informationen zu S3-in-Outposts-ARNs finden Sie unter [Ressource ARNs für S3 auf Outposts](#).

Note

Mit Amazon S3 on Outposts werden Objektdaten immer im Outpost gespeichert. Wenn AWS ein Outpost-Rack installiert, bleiben Ihre Daten in Ihrem Outpost lokal, um die Anforderungen an die Datenspeicherorte zu erfüllen. Ihre Objekte verlassen niemals Ihren Outpost und befinden sich nicht in einer AWS-Region. Da die AWS Management Console innerhalb der Region gehostet wird, können Sie die Konsole nicht verwenden, um Objekte in Ihrem Outpost hochzuladen oder zu verwalten. Sie können die REST-API, AWS Command Line Interface (AWS CLI) und AWS-SDKs zum Hochladen und Verwalten Ihrer Objekte über Ihre Zugriffspunkte verwenden.

Die folgenden Beispiele für AWS Command Line Interface (AWS CLI) und AWS SDK for Java veranschaulichen, wie Sie die API-Operation HeadBucket verwenden, um festzustellen, ob ein Amazon-S3-on-Outpost-Bucket vorhanden ist und ob Sie Zugriffsberechtigungen für diesen Bucket besitzen. Weitere Informationen finden Sie unter [HeadBucket](#) in der API-Referenz zu Amazon Simple Storage Service.

Verwendung der AWS CLI

Im folgenden AWS CLI-Beispiel für S3 on Outposts wird der Befehl `head-bucket` verwendet, um festzustellen, ob ein Bucket vorhanden ist und ob Sie Zugriffsberechtigungen für diesen Bucket besitzen. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [head-bucket](#) in der AWS CLI-Referenz.

```
aws s3api head-bucket --bucket arn:aws:s3-  
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-  
access-point
```

Verwenden des AWS-SDKs für Java

Das folgende S3-on-Outposts-Beispiel veranschaulicht, wie Sie feststellen, ob ein Bucket vorhanden ist und ob Sie Zugriffsberechtigungen für diesen Bucket besitzen. Wenn Sie dieses Beispiel verwenden möchten, geben Sie den Zugriffspunkt-ARN für den Outpost an. Weitere Informationen finden Sie unter [HeadBucket](#) in der API-Referenz zu Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.HeadBucketRequest;  
  
public class HeadBucket {  
    public static void main(String[] args) {  
        String accessPointArn = "*** access point ARN ***";  
  
        try {  
            // This code expects that you have AWS credentials set up per:  
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-  
credentials.html  
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()  
                .enableUseArnRegion()  
                .build();  
  
            s3Client.headBucket(new HeadBucketRequest(accessPointArn));  
        } catch (AmazonServiceException e) {  
            // The call was transmitted successfully, but Amazon S3 couldn't process  
            // it, so it returned an error response.  
            e.printStackTrace();  
        }  
    }  
}
```

```
    } catch (SdkClientException e) {  
        // Amazon S3 couldn't be contacted for a response, or the client  
        // couldn't parse the response from Amazon S3.  
        e.printStackTrace();  
    }  
}  
}
```

Durchführen und Verwalten eines mehrteiligen Uploads mit dem SDK for Java

Mit Amazon S3 in Outposts können Sie S3-Buckets in Ihren AWS Outposts-Ressourcen erstellen und Objekte On-Premises für Anwendungen speichern und abrufen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Die folgenden Beispiele veranschaulichen, wie Sie S3 on Outposts mit der AWS SDK for Java verwenden können, um einen mehrteiligen Upload durchzuführen und zu verwalten.

Themen

- [Durchführen eines mehrteiligen Uploads eines Objekts in einem S3-on-Outposts-Bucket](#)
- [Kopieren eines großen Objekts in einem S3-on-Outposts-Bucket mithilfe eines mehrteiligen Uploads](#)
- [Auflisten von Teilen eines Objekts in einem S3-on-Outposts-Bucket](#)
- [Abrufen einer Liste der in Bearbeitung befindlichen mehrteiligen Uploads in einem S3-on-Outposts-Bucket](#)

Durchführen eines mehrteiligen Uploads eines Objekts in einem S3-on-Outposts-Bucket

Das folgende S3-on-Outposts-Beispiel initiiert, lädt und beendet einen mehrteiligen Upload eines Objekts in einen Bucket mithilfe des SDK for Java. Zum Verwenden dieses Beispiels ersetzen Sie jeden *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen finden Sie unter [Hochladen eines Objekts mit Multipart-Upload](#).

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);
```

```
        // Copy this part.
        CopyPartRequest copyRequest = new CopyPartRequest()
            .withSourceBucketName(accessPointArn)
            .withSourceKey(sourceObjectKey)
            .withDestinationBucketName(accessPointArn)
            .withDestinationKey(destObjectKey)
            .withUploadId(initResult.getUploadId())
            .withFirstByte(bytePosition)
            .withLastByte(lastByte)
            .withPartNumber(partNum++);
        copyResponses.add(s3Client.copyPart(copyRequest));
        bytePosition += partSize;
    }

    // Complete the upload request to concatenate all uploaded parts and make
    // the copied object available.
    CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
        accessPointArn,
        destObjectKey,
        initResult.getUploadId(),
        getETags(copyResponses));
    s3Client.completeMultipartUpload(completeRequest);
    System.out.println("Multipart copy complete.");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

// This is a helper function to construct a list of ETags.
private static List<PartETag> getETags(List<CopyPartResult> responses) {
    List<PartETag> etags = new ArrayList<PartETag>();
    for (CopyPartResult response : responses) {
        etags.add(new PartETag(response.getPartNumber(), response.getETag()));
    }
    return etags;
}
```

Kopieren eines großen Objekts in einem S3-on-Outposts-Bucket mithilfe eines mehrteiligen Uploads

Im folgenden Beispiel wird ein Objekt mithilfe des SDK for Java in einem S3-on-Outposts-Bucket kopiert. Zum Verwenden dieses Beispiels ersetzen Sie jeden *user input placeholder* durch Ihre eigenen Informationen. Dieses Beispiel basiert auf [Kopieren eines Objekts mit Multipart-Upload](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();
```

```
// Copy the object using 5 MB parts.
long partSize = 5 * 1024 * 1024;
long bytePosition = 0;
int partNum = 1;
List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
while (bytePosition < objectSize) {
    // The last part might be smaller than partSize, so check to make sure
    // that lastByte isn't beyond the end of the object.
    long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

    // Copy this part.
    CopyPartRequest copyRequest = new CopyPartRequest()
        .withSourceBucketName(accessPointArn)
        .withSourceKey(sourceObjectKey)
        .withDestinationBucketName(accessPointArn)
        .withDestinationKey(destObjectKey)
        .withUploadId(initResult.getUploadId())
        .withFirstByte(bytePosition)
        .withLastByte(lastByte)
        .withPartNumber(partNum++);
    copyResponses.add(s3Client.copyPart(copyRequest));
    bytePosition += partSize;
}

// Complete the upload request to concatenate all uploaded parts and make
the copied object available.
CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
    accessPointArn,
    destObjectKey,
    initResult.getUploadId(),
    getETags(copyResponses));
s3Client.completeMultipartUpload(completeRequest);
System.out.println("Multipart copy complete.");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

```
// This is a helper function to construct a list of ETags.
private static List<PartETag> getETags(List<CopyPartResult> responses) {
    List<PartETag> etags = new ArrayList<PartETag>();
    for (CopyPartResult response : responses) {
        etags.add(new PartETag(response.getPartNumber(), response.getETag()));
    }
    return etags;
}
}
```

Auflisten von Teilen eines Objekts in einem S3-on-Outposts-Bucket

Das folgende S3-on-Outposts-Beispiel listet die Teile eines Objekts in einem Bucket mit dem SDK for Java auf. Zum Verwenden dieses Beispiels ersetzen Sie jeden *user input placeholder* durch Ihre eigenen Informationen.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.List;

public class ListParts {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** Key name ***";
        String uploadId = "*** Upload ID ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
            // credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            ListPartsRequest listPartsRequest = new ListPartsRequest(accessPointArn,
                keyName, uploadId);
            PartListing partListing = s3Client.listParts(listPartsRequest);
            List<PartSummary> partSummaries = partListing.getParts();
        }
    }
}
```

```

        System.out.println(partSummaries.size() + " multipart upload parts");
        for (PartSummary p : partSummaries) {
            System.out.println("Upload part: Part number = \"" + p.getPartNumber()
+ "\", ETag = " + p.getETag());
        }

    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
}

```

Abrufen einer Liste der in Bearbeitung befindlichen mehrteiligen Uploads in einem S3-on-Outposts-Bucket

Das folgende S3-on-Outposts-Beispiel zeigt, wie Sie mit dem SDK for Java eine Liste der laufenden mehrteiligen Uploads aus einem Outposts-Bucket abrufen. Zum Verwenden dieses Beispiels ersetzen Sie jeden *user input placeholder* durch Ihre eigenen Informationen. Dies ist ein Beispiel, das aus dem Beispiel für [Auflisten von mehrteiligen Uploads](#) für Amazon S3 adaptiert wurde.

```

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListMultipartUploadsRequest;
import com.amazonaws.services.s3.model.MultipartUpload;
import com.amazonaws.services.s3.model.MultipartUploadListing;

import java.util.List;

public class ListMultipartUploads {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {

```



```
// This code expects that you have AWS credentials set up per:
// https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
    .enableUseArnRegion()
    .build();

// Retrieve a list of all in-progress multipart uploads.
ListMultipartUploadsRequest allMultipartUploadsRequest = new
ListMultipartUploadsRequest(accessPointArn);
MultipartUploadListing multipartUploadListing =
s3Client.listMultipartUploads(allMultipartUploadsRequest);
List<MultipartUpload> uploads =
multipartUploadListing.getMultipartUploads();

// Display information about all in-progress multipart uploads.
System.out.println(uploads.size() + " multipart upload(s) in progress.");
for (MultipartUpload u : uploads) {
    System.out.println("Upload in progress: Key = \"" + u.getKey() + "\",
id = " + u.getUploadId());
}
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

Verwenden vorsignierter URLs für S3 on Outposts

Um zeitlich begrenzten Zugriff auf Objekte zu gewähren, die lokal auf einem Outpost gespeichert sind, ohne Ihre Bucket-Richtlinie zu aktualisieren, können Sie eine vordefinierte URL verwenden. Mit vorsignierten URLs können Sie als Bucket-Besitzer Objekte für Personen in Ihrer Virtual Private Cloud (VPC) freigeben oder ihnen die Möglichkeit geben, Objekte hochzuladen oder zu löschen.

Wenn Sie eine vorsignierte URL mit Hilfe der AWS-SDKs oder AWS Command Line Interface (AWS CLI) erstellen, verknüpfen Sie die URL mit einer bestimmten Aktion. Sie können auch einen

zeitlich begrenzten Zugriff auf die vorsignierte URL gewähren, indem Sie eine benutzerdefinierte Ablaufzeit wählen, die zwischen 1 Sekunde und 7 Tagen liegen kann. Wenn Sie die vorsignierte URL freigeben, kann die Person in der VPC die in der URL eingebettete Aktion so ausführen, als wäre sie der ursprünglich signierende Benutzer. Wenn die URL ihre Verfallszeit erreicht, läuft sie ab und funktioniert nicht mehr.

Beschränkung der Funktionen für vorsignierte URLs

Die Funktionen einer vorsignierten URL sind durch die Berechtigungen des Benutzers eingeschränkt, der sie erstellt hat. Im Wesentlichen sind vorsignierte URLs Inhaber-Token, die denjenigen, die sie besitzen, Zugriff gewähren. Daher empfehlen wir Ihnen, sie angemessen zu schützen.

AWS Signature Version 4 (SigV4)

Um ein bestimmtes Verhalten zu erzwingen, wenn vorsignierte URL-Anfragen mit AWS Signature Version 4 (SigV4) authentifiziert werden, können Sie Bedingungsschlüssel in Bucket-Richtlinien und Zugriffspunkt-Richtlinien verwenden. Sie können z. B. eine Bucket-Richtlinie erstellen, die die `s3-outposts:signatureAge`-Bedingung verwendet, um jede vorsignierte URL-Anfrage von Amazon S3 on Outposts für Objekte im `example-outpost-bucket`-Bucket zu verweigern, wenn die Signatur mehr als 10 Minuten alt ist. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Deny a presigned URL request if the signature is more than 10
minutes old",
      "Effect": "Deny",
      "Principal": {"AWS": "444455556666"},
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
      "Condition": {
        "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
        "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
      }
    }
  ]
}
```

Eine Liste von Bedingungsschlüsseln und zusätzlichen Beispielrichtlinien, die Sie verwenden können, um ein bestimmtes Verhalten zu erzwingen, wenn vorsignierte URL-Anfragen mit Hilfe von Signature Version 4 authentifiziert werden, finden Sie unter [AWS-Signature Version 4 \(SigV4\) – Authentifizierungsspezifische Richtlinienschlüssel](#).

Beschränkung der Netzwege

Wenn Sie die Verwendung von vorsignierten URLs und den Zugriff aller S3 on Outposts auf bestimmte Netzwerkpfade beschränken möchten, können Sie Richtlinien schreiben, die einen bestimmten Netzwerkpfad erfordern. Um die Beschränkung auf den IAM-Prinzipal festzulegen, der den Anruf tätigt, können Sie identitätsbasierte AWS Identity and Access Management (IAM)-Richtlinien verwenden (z. B. Benutzer-, Gruppen- oder Rollenrichtlinien). Um die Beschränkung für die Ressource S3 on Outposts festzulegen, können Sie ressourcenbasierte Richtlinien verwenden (z. B. Bucket- und Zugriffspunkt-Richtlinien).

Eine Netzwerkpfadbeschränkung für den IAM-Prinzipal erfordert, dass der Benutzer dieser Anmeldeinformationen Anfragen aus dem angegebenen Netzwerk stellt. Eine Einschränkung des Buckets oder des Zugriffspunkts erfordert, dass alle Anfragen an diese Ressource aus dem angegebenen Netz stammen. Diese Einschränkungen gelten auch außerhalb des Szenarios der vorsignierten URL.

Die globale IAM-Bedingung, die Sie verwenden, hängt von der Art des Endpunkts ab. Wenn Sie den öffentlichen Endpunkt für S3 on Outposts verwenden, benutzen Sie `aws:SourceIp`. Wenn Sie einen VPC-Endpunkt für S3 on Outposts verwenden, verwenden Sie `aws:SourceVpc` oder `aws:SourceVpce`.

Die folgende IAM-Richtlinienanweisung verlangt, dass der Prinzipal ausschließlich über den angegebenen Netzwerkbereich auf AWS zugreift. Mit dieser Richtlinie müssen alle Zugriffe von diesem Bereich ausgehen. Dies gilt auch für den Fall, dass jemand eine vorsignierte URL für S3 on Outposts verwendet. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
{
  "Sid": "NetworkRestrictionForIAMPrincipal",
  "Effect": "Deny",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
    "BoolIfExists": {"aws:ViaAWSService": "false"}
  }
}
```

```
}  
}
```

Ein Beispiel für eine Bucket-Richtlinie, die den `aws:SourceIP` AWS globalen Bedingungsschlüssel verwendet, um den Zugriff auf einen S3 on Outposts-Bucket auf einen bestimmten Netzwerkbereich zu beschränken, finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

Wer eine vorsignierte URL erstellen kann

Alle Benutzer mit gültigen Sicherheitsanmeldeinformationen können vorsignierte URLs erstellen. Damit ein Benutzer in der VPC jedoch erfolgreich auf ein Objekt zugreifen kann, muss die zugewiesene URL von jemandem erstellt werden, der die Berechtigung hat, den Vorgang durchzuführen, auf dem die zugewiesene URL basiert.

Sie können die folgenden Anmeldeinformationen verwenden, um eine vorsignierte URL zu erstellen:

- IAM-Instance-Profil – Bis zu 6 Stunden gültig.
- AWS Security Token Service – Gültig bis zu 36 Stunden, wenn mit dauerhaften Anmeldeinformationen signiert wird, z. B. mit den Anmeldeinformationen des AWS-Konto Stammbenutzers oder eines IAM-Benutzers.
- IAM-Benutzer - Gültig bis zu 7 Tage, wenn Sie die AWS Signature Version 4 verwenden.

Um eine vordefinierte URL zu erstellen, die bis zu 7 Tage gültig ist, delegieren Sie zunächst die IAM-Benutzer-Anmeldeinformationen (den Zugriffsschlüssel und den geheimen Schlüssel) an das von Ihnen verwendete SDK. Erzeugen Sie dann eine vorsignierte URL, indem Sie AWS Signature Version 4 verwenden.

Note

- Wenn Sie eine vorsignierte URL mit einem temporären Token erstellt haben, läuft die URL ab, wenn das Token abläuft, auch wenn Sie die URL mit einer späteren Ablaufzeit erstellt haben.
- Da vorsignierte URLs jedem, der über die URL verfügt, Zugriff auf Ihre S3 on Outposts-Buckets gewähren, empfehlen wir Ihnen, diese entsprechend zu schützen. Weitere Informationen zum Schutz von vorsignierten URLs finden Sie unter [Beschränkung der Funktionen für vorsignierte URLs](#).

Wann prüft S3 on Outposts das Ablaufdatum und die Uhrzeit einer vorsignierten URL?

Zum Zeitpunkt der HTTP-Anfrage überprüft S3 on Outposts das Ablaufdatum und die Uhrzeit einer signierten URL. Beginnt ein Client beispielsweise mit dem Herunterladen einer großen Datei unmittelbar vor der Ablaufzeit, wird der Download auch dann fortgesetzt, wenn die Ablaufzeit während des Downloads verstreicht. Wenn die Verbindung jedoch unterbrochen wird und der Client versucht, den Download nach Ablauf der Zeit erneut zu starten, schlägt der Download fehl.

Weitere Informationen zur Verwendung einer vorsignierten URL zum Teilen oder Hochladen von Objekten finden Sie in den folgenden Themen.

Themen

- [Gemeinsame Nutzung von Objekten unter Verwendung vorsignierter URLs](#)
- [Generierung einer vorsignierten URL zum Hochladen eines Objekts in einen S3 on Outposts-Bucket](#)

Gemeinsame Nutzung von Objekten unter Verwendung vorsignierter URLs

Um zeitlich begrenzten Zugriff auf Objekte zu gewähren, die lokal auf einem Outpost gespeichert sind, ohne Ihre Bucket-Richtlinie zu aktualisieren, können Sie eine vorsignierte URL verwenden. Mit vorsignierten URLs können Sie als Bucket-Besitzer Objekte für Personen in Ihrer Virtual Private Cloud (VPC) freigeben oder ihnen die Möglichkeit geben, Objekte hochzuladen oder zu löschen.

Wenn Sie eine vorsignierte URL mithilfe der AWS SDKs oder der AWS Command Line Interface (AWS CLI) erstellen, verknüpfen Sie die URL mit einer bestimmten Aktion. Sie können auch einen zeitlich begrenzten Zugriff auf die vorsignierte URL gewähren, indem Sie eine benutzerdefinierte Ablaufzeit wählen, die zwischen 1 Sekunde und 7 Tagen liegen kann. Wenn Sie die vorsignierte URL freigeben, kann die Person in der VPC die in der URL eingebettete Aktion so ausführen, als wäre sie der ursprünglich signierende Benutzer. Wenn die URL ihre Verfallszeit erreicht, läuft sie ab und funktioniert nicht mehr.

Wenn Sie eine vorsignierte URL erstellen, müssen Sie Ihre Sicherheitsanmeldedaten eingeben und dann Folgendes angeben:

- Ein Zugriffspunkt Amazon-Ressourcenname (ARN) für den Amazon S3 on Outposts Bucket
- Ein Objektschlüssel
- Eine HTTP-Methode (GET zum Herunterladen von Objekten)

- Ein Verfallsdatum und eine Verfallszeit

Eine vorsignierte URL ist nur für die angegebene Dauer gültig. Das heißt, Sie müssen die von der URL erlaubte Aktion vor dem Ablaufdatum und der Ablaufzeit starten. Sie können eine vorsignierte URL bis zum Ablaufdatum und zur Ablaufzeit mehrfach verwenden. Wenn Sie eine vorsignierte URL mit einem temporären Token erstellt haben, läuft die URL ab, wenn das Token abläuft, auch wenn Sie die URL mit einer späteren Ablaufzeit erstellt haben.

Benutzer in der Virtual Private Cloud (VPC), die Zugriff auf die vorsignierte URL haben, können auf das Objekt zugreifen. Wenn Sie beispielsweise ein Video in Ihrem Bucket haben und sowohl der Bucket als auch das Objekt privat sind, können Sie das Video mit anderen teilen, indem Sie eine vorsignierte URL generieren. Da vorsignierte URLs jedem, der über die URL verfügt, Zugriff auf Ihre S3 on Outposts-Buckets gewähren, empfehlen wir Ihnen, diese URLs entsprechend zu schützen. Weitere Informationen zum Schutz vorsignierter URLs finden Sie unter [Beschränkung der Funktionen für vorsignierte URLs](#).

Alle Benutzer mit gültigen Sicherheitsanmeldeinformationen können vorsignierte URLs erstellen. Die vorsignierte URL muss jedoch von jemandem erstellt werden, der die Berechtigung hat, den Vorgang durchzuführen, auf dem die vorsignierte URL basiert. Weitere Informationen finden Sie unter [Wer eine vorsignierte URL erstellen kann](#).

Sie können eine vorsignierte URL zur Freigabe eines Objekts in einem S3 on Outposts-Bucket generieren, indem Sie die AWS -SDKs und die AWS CLI anwenden. Weitere Informationen finden Sie in den folgenden Beispielen.

Verwenden der SDKs AWS

Sie können die AWS SDKs verwenden, um eine vorsignierte URL zu generieren, die Sie an andere weitergeben können, damit diese ein Objekt abrufen können.

Note

Wenn Sie die AWS SDKs verwenden, um eine vorsignierte URL zu generieren, beträgt die maximale Ablaufzeit für eine vorsignierte URL 7 Tage ab dem Zeitpunkt der Erstellung.

Java

Example

Das folgende Beispiel generiert eine vorsignierte URL, die Sie an andere weitergeben können, damit diese ein Objekt aus einem S3 on Outposts-Bucket abrufen können. Weitere Informationen finden Sie unter [Verwenden vorsignierter URLs für S3 on Outposts](#). Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter [Erste Schritte im AWS SDK for Java Entwicklerhandbuch](#).

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.HttpMethod;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GeneratePresignedUrlRequest;

import java.io.IOException;
import java.net.URL;
import java.time.Instant;

public class GeneratePresignedURL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String accessPointArn = "*** access point ARN ***";
        String objectKey = "*** object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .build();

            // Set the presigned URL to expire after one hour.
            java.util.Date expiration = new java.util.Date();
            long expTimeMillis = Instant.now().toEpochMilli();
            expTimeMillis += 1000 * 60 * 60;
            expiration.setTime(expTimeMillis);
```

```
// Generate the presigned URL.
System.out.println("Generating pre-signed URL.");
GeneratePresignedUrlRequest generatePresignedUrlRequest =
    new GeneratePresignedUrlRequest(accessPointArn, objectKey)
        .withMethod(HttpMethod.GET)
        .withExpiration(expiration);
URL url = s3Client.generatePresignedUrl(generatePresignedUrlRequest);

System.out.println("Pre-Signed URL: " + url.toString());
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't
process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
```

.NET

Example

Das folgende Beispiel generiert eine vorsignierte URL, die Sie an andere weitergeben können, damit diese ein Objekt aus einem S3 on Outposts-Bucket abrufen können. Weitere Informationen finden Sie unter [Verwenden vorsignierter URLs für S3 on Outposts](#). Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter [Getting Started with the AWS SDK for .NET](#) im AWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;

namespace Amazon.DocSamples.S3
{
    class GenPresignedURLTest
```



```
{
    private const string accessPointArn = "*** access point ARN ***";
    private const string objectKey = "*** object key ***";
    // Specify how long the presigned URL lasts, in hours.
    private const double timeoutDuration = 12;
    // Specify your bucket Region (an example Region is shown).
    private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
    private static IAmazonS3 s3Client;

    public static void Main()
    {
        s3Client = new AmazonS3Client(bucketRegion);
        string urlString = GeneratePreSignedURL(timeoutDuration);
    }
    static string GeneratePreSignedURL(double duration)
    {
        string urlString = "";
        try
        {
            GetPreSignedUrlRequest request1 = new GetPreSignedUrlRequest
            {
                BucketName = accessPointArn,
                Key = objectKey,
                Expires = DateTime.UtcNow.AddHours(duration)
            };
            urlString = s3Client.GetPreSignedURL(request1);
        }
        catch (AmazonS3Exception e)
        {
            Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
        }
        return urlString;
    }
}
}
```

Python

Das folgende Beispiel generiert eine vorsignierte URL zur Freigabe eines Objekts mit Hilfe des SDK für Python (Boto3). Verwenden Sie z. B. einen Boto3-Client und die `generate_presigned_url` Funktion, um eine vorsignierte URL zu generieren, die Ihnen ermöglicht zu GET ein Objekt.

```
import boto3
url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'ACCESS_POINT_ARN', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

Weitere Informationen zur Verwendung von SDK for Python (Boto3) zur Erzeugung einer vorsignierten URL finden Sie unter [Python](#) in der API-Referenz für AWS SDK for Python (Boto) .

Verwenden von AWS CLI

Der folgende AWS CLI Beispielbefehl generiert eine vorsignierte URL für einen S3 on Outposts-Bucket. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Note

Wenn Sie den verwenden, AWS CLI um eine vorsignierte URL zu generieren, beträgt die maximale Ablaufzeit für eine vorsignierte URL 7 Tage ab dem Zeitpunkt der Erstellung.

```
aws s3 presign s3://arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-
point/mydoc.txt --expires-in 604800
```

Weitere Informationen finden Sie unter [vorsignieren](#) in der AWS CLI Befehlsreferenz.

Generierung einer vorsignierten URL zum Hochladen eines Objekts in einen S3 on Outposts-Bucket

Um zeitlich begrenzten Zugriff auf Objekte zu gewähren, die lokal auf einem Outpost gespeichert sind, ohne Ihre Bucket-Richtlinie zu aktualisieren, können Sie eine vorsignierte URL verwenden.

Mit vorsignierten URLs können Sie als Bucket-Besitzer Objekte für Personen in Ihrer Virtual Private Cloud (VPC) freigeben oder ihnen die Möglichkeit geben, Objekte hochzuladen oder zu löschen.

Wenn Sie eine vorsignierte URL mit Hilfe der AWS-SDKs oder AWS Command Line Interface (AWS CLI) erstellen, verknüpfen Sie die URL mit einer bestimmten Aktion. Sie können auch einen zeitlich begrenzten Zugriff auf die vorsignierte URL gewähren, indem Sie eine benutzerdefinierte Ablaufzeit wählen, die zwischen 1 Sekunde und 7 Tagen liegen kann. Wenn Sie die vorsignierte URL freigeben, kann die Person in der VPC die in der URL eingebettete Aktion so ausführen, als wäre sie der ursprünglich signierende Benutzer. Wenn die URL ihre Verfallszeit erreicht, läuft sie ab und funktioniert nicht mehr.

Wenn Sie eine vorsignierte URL erstellen, müssen Sie Ihre Sicherheitsanmeldedaten eingeben und dann Folgendes angeben:

- Ein Zugriffspunkt Amazon-Ressourcenname (ARN) für den Amazon S3 on Outposts Bucket
- Ein Objektschlüssel
- Eine HTTP-Methode (PUT zum Hochladen von Objekten)
- Ein Verfallsdatum und eine Verfallszeit

Eine vorsignierte URL ist nur für die angegebene Dauer gültig. Das heißt, Sie müssen die von der URL erlaubte Aktion vor dem Ablaufdatum und der Ablaufzeit starten. Sie können eine vorsignierte URL bis zum Ablaufdatum und zur Ablaufzeit mehrfach verwenden. Wenn Sie eine vorsignierte URL mit einem temporären Token erstellt haben, läuft die URL ab, wenn das Token abläuft, auch wenn Sie die URL mit einer späteren Ablaufzeit erstellt haben.

Wenn die von einer vorsignierten URL erlaubte Aktion aus mehreren Schritten besteht, wie z. B. ein mehrteiliger Upload, müssen Sie alle Schritte vor Ablauf der Zeit starten. Wenn S3 on Outposts versucht, einen Schritt mit einer abgelaufenen URL zu starten, erhalten Sie eine Fehlermeldung.

Benutzer in der Virtual Private Cloud (VPC), die Zugriff auf die vorsignierte URL haben, können Objekte hochladen. So kann beispielsweise ein Benutzer in der VPC, der Zugriff auf die vorsignierte URL hat, ein Objekt in Ihren Bucket hochladen. Da vorsignierte URLs jedem Benutzer in der VPC, der Zugriff auf die vorsignierte URL hat, Zugriff auf Ihren S3 on Outposts-Bucket gewähren, empfehlen wir Ihnen, diese URLs entsprechend zu schützen. Weitere Informationen zum Schutz vorsignierter URLs finden Sie unter [Beschränkung der Funktionen für vorsignierte URLs](#).

Alle Benutzer mit gültigen Sicherheitsanmeldeinformationen können vorsignierte URLs erstellen. Die vorsignierte URL muss jedoch von jemandem erstellt werden, der die Berechtigung hat, den Vorgang

durchzuführen, auf dem die vorsignierte URL basiert. Weitere Informationen finden Sie unter [Wer eine vorsignierte URL erstellen kann](#).

Verwendung der AWS-SDKs zur Generierung einer vorsignierten URL für eine S3 on Outposts-Objektoperation

Java

SDK für Java 2.x

Dieses Beispiel zeigt, wie Sie eine vorsignierte URL generieren, mit der Sie ein Objekt für eine begrenzte Zeit in einen S3 on Outposts-Bucket hochladen können. Weitere Informationen finden Sie unter [Verwenden vorsignierter URLs für S3 on Outposts](#).

```
public static void signBucket(S3Presigner presigner, String
outpostAccessPointArn, String keyName) {

    try {
        PutObjectRequest objectRequest = PutObjectRequest.builder()
            .bucket(accessPointArn)
            .key(keyName)
            .contentType("text/plain")
            .build();

        PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
            .signatureDuration(Duration.ofMinutes(10))
            .putObjectRequest(objectRequest)
            .build();

        PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);

        String myURL = presignedRequest.url().toString();
        System.out.println("Presigned URL to upload a file to: " +myURL);
        System.out.println("Which HTTP method must be used when uploading a
file: " +
            presignedRequest.httpRequest().method());

        // Upload content to the S3 on Outposts bucket by using this URL.
        URL url = presignedRequest.url();
```

```
        // Create the connection and use it to upload the new object by using
the presigned URL.
        HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
        connection.setDoOutput(true);
        connection.setRequestProperty("Content-Type","text/plain");
        connection.setRequestMethod("PUT");
        OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream());
        out.write("This text was uploaded as an object by using a presigned
URL.");
        out.close();

        connection.getResponseCode();
        System.out.println("HTTP response code is " +
connection.getResponseCode());

    } catch (S3Exception e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Python

SDK für Python (Boto3)

In diesem Beispiel wird gezeigt, wie man eine vorsignierte URL generiert, die für eine begrenzte Zeit eine S3 on Outposts-Aktion ausführen kann. Weitere Informationen finden Sie unter [Verwenden vorsignierter URLs für S3 on Outposts](#). Um eine Anfrage mit der URL zu stellen, verwenden Sie das Requests Paket.

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)
```

```
def generate_presigned_url(s3_client, client_method, method_parameters,
                           expires_in):
    """
    Generate a presigned S3 on Outposts URL that can be used to perform an
    action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds that the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
        logger.info("Got presigned URL: %s", url)
    except ClientError:
        logger.exception(
            "Couldn't get a presigned URL for client method '%s'.",
            client_method)
        raise
    return url

def usage_demo():
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

    print('-'*88)
    print("Welcome to the Amazon S3 on Outposts presigned URL demo.")
    print('-'*88)

    parser = argparse.ArgumentParser()
    parser.add_argument('accessPointArn', help="The name of the S3 on Outposts
    access point ARN.")
    parser.add_argument(
        'key', help="For a GET operation, the key of the object in S3 on
    Outposts. For a "
        "PUT operation, the name of a file to upload.")
    parser.add_argument(
        'action', choices=('get', 'put'), help="The action to perform.")
    args = parser.parse_args()
```

```
s3_client = boto3.client('s3')
client_action = 'get_object' if args.action == 'get' else 'put_object'
url = generate_presigned_url(
    s3_client, client_action, {'Bucket': args.accessPointArn, 'Key':
args.key}, 1000)

print("Using the Requests package to send a request to the URL.")
response = None
if args.action == 'get':
    response = requests.get(url)
elif args.action == 'put':
    print("Putting data to the URL.")
    try:
        with open(args.key, 'r') as object_file:
            object_text = object_file.read()
            response = requests.put(url, data=object_text)
    except FileNotFoundError:
        print(f"Couldn't find {args.key}. For a PUT operation, the key must
be the "
            f"name of a file that exists on your computer.")

if response is not None:
    print("Got response:")
    print(f"Status: {response.status_code}")
    print(response.text)

print('-'*88)

if __name__ == '__main__':
    usage_demo()
```

Amazon S3 on Outposts mit lokalem Amazon EMR on Outposts

Amazon EMR ist eine verwaltete Cluster-Plattform, die die Ausführung von Big-Data-Frameworks wie Apache Hadoop und vereinfacht Apache Spark, AWS um riesige Datenmengen zu verarbeiten und zu analysieren. Durch die Verwendung dieser Frameworks und verwandter Open-Source-Projekte können Sie Daten für Analysezwecke und Business-Intelligence-Workloads verarbeiten. Amazon EMR unterstützt Sie auch bei der Transformation und Übertragung großer Datenmengen in und aus anderen AWS Datenspeichern und Datenbanken und unterstützt Amazon S3 on Outposts. Weitere

Informationen zu Amazon EMR finden Sie unter [Amazon EMR in Outposts](#) im Verwaltungshandbuch für Amazon EMR.

Für Amazon S3 on Outposts begann Amazon EMR mit der Unterstützung des Apache Hadoop S3A-Konnektors in Version 7.0.0. Frühere Versionen von Amazon EMR unterstützen lokale S3 on Outposts nicht, und das EMR File System (EMRFS) wird nicht unterstützt.

Unterstützte Anwendungen

Amazon EMR mit Amazon S3 on Outposts unterstützt die folgenden Anwendungen:

- Hadoop
- Spark
- Hue
- Hive
- Sqoop
- Pig
- Hudi
- Flink

Weitere Informationen finden Sie im [Handbuch zu Amazon-EMR-Versionen](#).

Erstellen und Konfigurieren eines Amazon S3-on-Outposts-Buckets

Amazon EMR verwendet die AWS SDK for Java mit Amazon S3 on Outposts, um Eingabe- und Ausgabedaten zu speichern. Ihre Amazon-EMR-Protokolldateien werden an einem von Ihnen ausgewählten regionalen Amazon S3-Speicherort gespeichert und nicht lokal auf dem Outpost gespeichert. Weitere Informationen finden Sie unter [Amazon-EMR-Protokolle](#) im Verwaltungshandbuch für Amazon EMR.

Um den Amazon S3- und DNS-Anforderungen zu entsprechen, haben S3-on-Outposts-Buckets Namensbeschränkungen und Einschränkungen. Weitere Informationen finden Sie unter [Erstellen eines S3-on-Outposts-Buckets](#).

Mit Amazon-EMR-Version 7.0.0 und höher können Sie Amazon EMR mit S3 on Outposts und dem S3A-Dateisystem verwenden.

Voraussetzungen

Berechtigungen für S3 on Outposts – Wenn Sie Ihr Amazon-EMR-Instance-Profil erstellen, muss Ihre Rolle den AWS Identity and Access Management (IAM)-Namespace für S3 on Outposts enthalten. S3 on Outposts hat seinen eigenen Namespace, `s3-outposts*`. Eine Beispielrichtlinie, die diesen Namespace verwendet, finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

S3A-Konnektor – Um Ihren EMR-Cluster für den Zugriff auf Daten aus einem Amazon S3-on-Outposts-Bucket zu konfigurieren, müssen Sie den Apache Hadoop S3A-Konnektor verwenden. Um den Konnektor zu verwenden, stellen Sie sicher, dass alle Ihre S3-URIs das `s3a` Schema verwenden. Andernfalls können Sie die Dateisystemimplementierung konfigurieren, die Sie für Ihren EMR-Cluster verwenden, sodass Ihre S3-URIs mit dem S3A-Konnektor funktionieren.

Um die Dateisystemimplementierung für die Arbeit mit dem S3A-Konnektor zu konfigurieren, verwenden Sie die `fs.AbstractFileSystem.file_scheme.impl` Konfigurationseigenschaften `fs.file_scheme.impl` und für Ihren EMR-Cluster, wobei dem Typ der S3-URIs `file_scheme` entspricht, die Sie haben. Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen. Um beispielsweise die Dateisystemimplementierung für S3-URIs zu ändern, die das `-s3` Schema verwenden, geben Sie die folgenden Cluster-Konfigurationseigenschaften an:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Um S3A zu verwenden, setzen Sie die `fs.file_scheme.impl` Konfigurationseigenschaft auf `org.apache.hadoop.fs.s3a.S3AFileSystem` und die `fs.AbstractFileSystem.file_scheme.impl` Eigenschaft auf `org.apache.hadoop.fs.s3a.S3A`.

Wenn Sie beispielsweise auf den Pfad zugreifen `s3a://bucket/...`, setzen Sie die `-fs.s3a.impl` Eigenschaft auf `org.apache.hadoop.fs.s3a.S3AFileSystem` und die `-fs.AbstractFileSystem.s3a.impl` Eigenschaft auf `org.apache.hadoop.fs.s3a.S3A`.

Erste Schritte mit Amazon EMR mit Amazon S3 on Outposts

In den folgenden Themen werden die ersten Schritte bei der Verwendung von Amazon EMR mit Amazon S3 on Outposts erläutert.

Themen

- [Erstellen einer Berechtigungsrichtlinie](#)
- [Ihren Cluster erstellen und konfigurieren](#)
- [Konfigurationsübersicht](#)
- [Überlegungen](#)

Erstellen einer Berechtigungsrichtlinie

Bevor Sie einen EMR-Cluster erstellen können, der Amazon S3 on Outposts verwendet, müssen Sie eine IAM-Richtlinie erstellen, die an das Amazon EC2-Instance-Profil für den Cluster angehängt wird. Die Richtlinie muss über Berechtigungen für den Zugriff auf den Amazon-Ressourcennamen (ARN) des S3-on-Outposts-Zugriffspunkts verfügen. Weitere Informationen zum Erstellen von IAM-Richtlinien für S3 on Outposts finden Sie unter [Einrichtung IAM mit S3 auf Outposts](#).

Die folgende Beispielrichtlinie zeigt, wie Sie die erforderlichen Berechtigungen erteilen. Nachdem Sie die Richtlinie erstellt haben, ordnen Sie die Richtlinie der Instance-Profilrolle zu, mit der Sie Ihren EMR-Cluster erstellen, wie im Abschnitt [the section called "Ihren Cluster erstellen und konfigurieren"](#) beschrieben. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name",
      "Action": [
        "s3-outposts:*"
      ]
    }
  ]
}
```

Ihren Cluster erstellen und konfigurieren

Führen Sie die folgenden Schritte in der Konsole aus, um einen Cluster zu erstellen, auf dem Spark mit S3 on Outposts ausgeführt wird.

So erstellen Sie einen Cluster, der Spark mit S3 on Outposts ausgeführt wird

1. Öffnen Sie die Amazon-EMR-Konsole unter <https://console.aws.amazon.com/elasticmapreduce/>.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie Cluster erstellen.
4. Wählen Sie für Amazon-EMR-Version emr-7.0.0 oder höher aus.
5. Wählen Sie für Anwendungspaket die Option Interaktive Spark- aus. Wählen Sie dann alle anderen unterstützten Anwendungen aus, die in Ihren Cluster aufgenommen werden sollen.
6. Um Amazon S3 on Outposts zu aktivieren, geben Sie Ihre Konfigurationseinstellungen ein.

Beispielkonfigurationseinstellungen

Um die folgenden Beispielkonfigurationseinstellungen zu verwenden, ersetzen Sie durch *user input placeholders* Ihre eigenen Informationen.

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3a.bucket.DOC-EXAMPLE-BUCKET.accesspoint.arn": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name"
      "fs.s3a.committer.name": "magic",
      "fs.s3a.select.enabled": "false"
    }
  },
  {
    "Classification": "hadoop-env",
    "Configurations": [
      {
        "Classification": "export",
        "Properties": {
          "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
        }
      }
    ]
  }
],
```

```
"Properties": {}
},
{
  "Classification": "spark-env",
  "Configurations": [
    {
      "Classification": "export",
      "Properties": {
        "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
      }
    }
  ],
  "Properties": {}
},
{
  "Classification": "spark-defaults",
  "Properties": {
    "spark.executorEnv.JAVA_HOME": "/usr/lib/jvm/java-11-amazon-
corretto.x86_64",
    "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
  }
}
]
```

7. Wählen Sie im Abschnitt Netzwerk eine Virtual Private Cloud (VPC) und ein Subnetz aus, die sich in Ihrem AWS Outposts Rack befinden. Weitere Informationen zu Amazon EMR in Outposts finden Sie unter [EMR-Cluster in AWS Outposts](#) im Verwaltungshandbuch für Amazon EMR.
8. Wählen Sie im Abschnitt EC2-Instance-Profil für Amazon EMR die IAM-Rolle aus, der [die zuvor erstellte Berechtigungsrichtlinie](#) angefügt ist.
9. Konfigurieren Sie Ihre verbleibenden Cluster-Einstellungen und wählen Sie dann Cluster erstellen aus.

Konfigurationsübersicht

In den folgenden Tabellen werden die S3A- und -SparkKonfigurationen sowie die Werte beschrieben, die für ihre Parameter angegeben werden müssen, wenn Sie einen Cluster einrichten, der S3 on Outposts mit Amazon EMR verwendet.

S3A-Konfigurationen

Parameter	Standardwert	Erforderlicher Wert für S3 on Outposts	Erklärung
<code>fs.s3a.aws.credentials.provider</code>	Wenn nicht angegeben, sucht S3A nach S3 im Regions-Bucket mit dem Namen des Outposts-Buckets.	Der Zugriffspunkt-ARN des S3-on-Outposts-Buckets	Amazon S3 on Outposts unterstützt reine Virtual-Private-Cloud(VPC)-Zugriffspunkte als einzige Möglichkeit, auf Ihre Outposts-Buckets zuzugreifen.
<code>fs.s3a.committer.name</code>	<code>file</code>	<code>magic</code>	Magic Committer ist der einzige unterstützte Committer für S3 on Outposts.
<code>fs.s3a.select.enabled</code>	TRUE	FALSE	S3 Select wird auf Outposts nicht unterstützt.
<code>JAVA_HOME</code>	<code>/usr/lib/jvm/java-8</code>	<code>/usr/lib/jvm/java-11-amazon-corretto.x86_64</code>	S3 on Outposts auf S3A erfordert Java Version 11.

Spark-Konfigurationen

Parameter	Standardwert	Erforderlicher Wert für S3 on Outposts	Erklärung
<code>spark.sql.sources.fastS3PartitionDiscovery.enabled</code>	TRUE	FALSE	S3 on Outposts unterstützt keine schnelle Partitionierung.

Parameter	Standardwert	Erforderlicher Wert für S3 on Outposts	Erklärung
spark.executorEnv.JAVA_HOME	/usr/lib/jvm/java-8	/usr/lib/jvm/java-11-amazon-corretto.x86_64	S3 on Outposts auf S3A erfordert Java-Version 11.

Überlegungen

Beachten Sie Folgendes, wenn Sie Amazon EMR in S3-on-Outposts-Buckets integrieren:

- Amazon S3 on Outposts wird mit Amazon-EMR-Version 7.0.0 und höher unterstützt.
- Der S3A-Konnektor ist erforderlich, um S3 on Outposts mit Amazon EMR zu verwenden. Nur S3A verfügt über die Funktionen, die für die Interaktion mit S3-on-Outposts-Buckets erforderlich sind. Informationen zur Einrichtung des S3A-[Konnektors finden Sie unter Voraussetzungen](#).
- Amazon S3 on Outposts unterstützt nur die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) mit Amazon EMR. Weitere Informationen finden Sie unter [the section called "Datenverschlüsselung"](#).
- Amazon S3 on Outposts unterstützt keine Schreibvorgänge mit dem S3A FileOutputCommitter. Schreibvorgänge mit den S3A FileOutputCommitter on S3 on Outposts-Buckets führen zu dem folgenden Fehler: InvalidStorageClass: Die von Ihnen angegebene Speicherklasse ist ungültig.
- Amazon S3 on Outposts wird mit Amazon EMR Serverless oder Amazon EMR in EKS nicht unterstützt.
- Amazon-EMR-Protokolle werden an einem von Ihnen ausgewählten regionalen Amazon S3-Speicherort gespeichert und nicht lokal im S3-on-Outposts-Bucket gespeichert.

Zwischenspeicherung von Autorisierung und Authentifizierung

S3 on Outposts speichert Authentifizierungs- und Autorisierungsdaten sicher lokal auf Outposts-Racks zwischen. Der Cache entfernt AWS-Region für jede Anforderung Roundtrips zum übergeordneten . Dadurch entfällt die Variabilität, die durch Netzwerk-Roundtrips eingeführt wird. Mit dem Authentifizierungs- und Autorisierungs-Cache in S3 on Outposts erhalten Sie konsistente Latenzen, die unabhängig von der Latenz der Verbindung zwischen den Outposts und der sind AWS-Region.

Wenn Sie eine S3-on-Outposts-API-Anfrage stellen, werden die Authentifizierungs- und Autorisierungsdaten sicher zwischengespeichert. Die zwischengespeicherten Daten werden dann verwendet, um nachfolgende S3-Objekt-API-Anforderungen zu authentifizieren. S3 on Outposts speichert Authentifizierungs- und Autorisierungsdaten nur zwischen, wenn die Anforderung mit Signature Version 4A (SigV4A) signiert wird. Der Cache wird lokal auf den Outposts innerhalb des S3-on-Outposts-Service gespeichert. Es wird asynchron aktualisiert, wenn Sie eine S3-API-Anfrage stellen. Der Cache ist verschlüsselt und es werden keine kryptografischen Klartextschlüssel auf Outposts gespeichert.

Der Cache ist bis zu 10 Minuten gültig, wenn der Outpost mit dem verbunden ist AWS-Region. Sie wird asynchron aktualisiert, wenn Sie eine S3-on-Outposts-API-Anforderung stellen, um sicherzustellen, dass die neuesten Richtlinien verwendet werden. Wenn der Outpost vom getrennt wird AWS-Region, ist der Cache bis zu 12 Stunden gültig.

Konfigurieren des Autorisierungs- und Authentifizierungs-Cache

S3 on Outposts speichert automatisch Authentifizierungs- und Autorisierungsdaten für Anfragen zwischen, die mit dem SigV4A-Algorithmus signiert sind. Weitere Informationen finden Sie unter [Signieren von AWS API-Anforderungen](#) im AWS Identity and Access Management - Benutzerhandbuch. Der SigV4A-Algorithmus ist in den neuesten Versionen der - AWS SDKs verfügbar. Sie können sie über eine Abhängigkeit von den [AWS Common Runtime \(CRT\)-Bibliotheken](#) abrufen.

Sie müssen die neueste Version des AWS SDK verwenden und die neueste Version des CRT installieren. Sie können beispielsweise ausführen, `pip install awscrt` um die neueste Version des CRT mit Boto3 zu erhalten.

S3 on Outposts speichert keine Authentifizierungs- und Autorisierungsdaten für Anfragen zwischen, die mit dem SigV4-Algorithmus signiert sind.

Validieren der SigV4A-Signatur

Sie können verwenden, AWS CloudTrail um zu überprüfen, ob Anfragen mit SigV4A signiert wurden. Weitere Informationen zum Einrichten von CloudTrail für S3 on Outposts finden Sie unter [Überwachen von S3 in Outposts mit Protokollen in AWS CloudTrail](#).

Nachdem Sie konfiguriert haben CloudTrail, können Sie überprüfen, wie eine Anforderung im `SignatureVersion` Feld der CloudTrail Protokolle signiert wurde. Für Anforderungen, die mit SigV4A signiert wurden, ist auf `SignatureVersion` festgelegt `AWS_4-ECDSA-P256-SHA256`.

Anforderungen, die mit SigV4 signiert wurden, haben auf `SignatureVersion` gesetzt `AWS 4-HMAC-SHA256`.

Sicherheit in S3 on Outposts

Die Sicherheit in der Cloud hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud selbst – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für Amazon S3 on Outposts gelten, finden Sie unter [Im Rahmen des Compliance-Programms zugelassene AWS-Services](#).
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, einschließlich der Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation erläutert, wie das Modell der geteilten Verantwortung bei der Verwendung von S3 on Outposts zum Tragen kommt. Die folgenden Themen veranschaulichen, wie Sie S3 on Outposts zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren außerdem, wie Sie andere AWS-Services verwenden, um Ihre S3-on-Outposts-Ressourcen zu überwachen und zu schützen.

Themen

- [Datenverschlüsselung in S3 on Outposts](#)
- [AWS PrivateLink für S3 auf Outposts](#)
- [AWS-Signature Version 4 \(SigV4\) – Authentifizierungsspezifische Richtlinienschlüssel](#)
- [AWS-verwaltete Richtlinien für Amazon S3 on Outposts](#)
- [Verwenden von serviceverknüpften Rollen für S3 on Outposts](#)

Datenverschlüsselung in S3 on Outposts

Standardmäßig werden alle in Amazon S3 on Outposts gespeicherten Daten mit serverseitiger Verschlüsselung über von Amazon S3 verwaltete Verschlüsselungsschlüssel (SSE-S3) verschlüsselt. Weitere Informationen finden Sie unter [Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden \(SSE-S3\)](#).

Sie können serverseitige Verschlüsselung optional mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verwenden. Wenn Sie SSE-C verwenden möchten, geben Sie einen Verschlüsselungsschlüssel als Teil Ihrer Objekt-API-Anforderungen an. Die serverseitige Verschlüsselung verschlüsselt nur die Objektdaten, nicht die Metadaten des Objekts. Weitere Informationen finden Sie unter [Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln \(-C\) SSE](#).

Note

S3 on Outposts unterstützt keine serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS)-Schlüsseln (SSE-KMS).

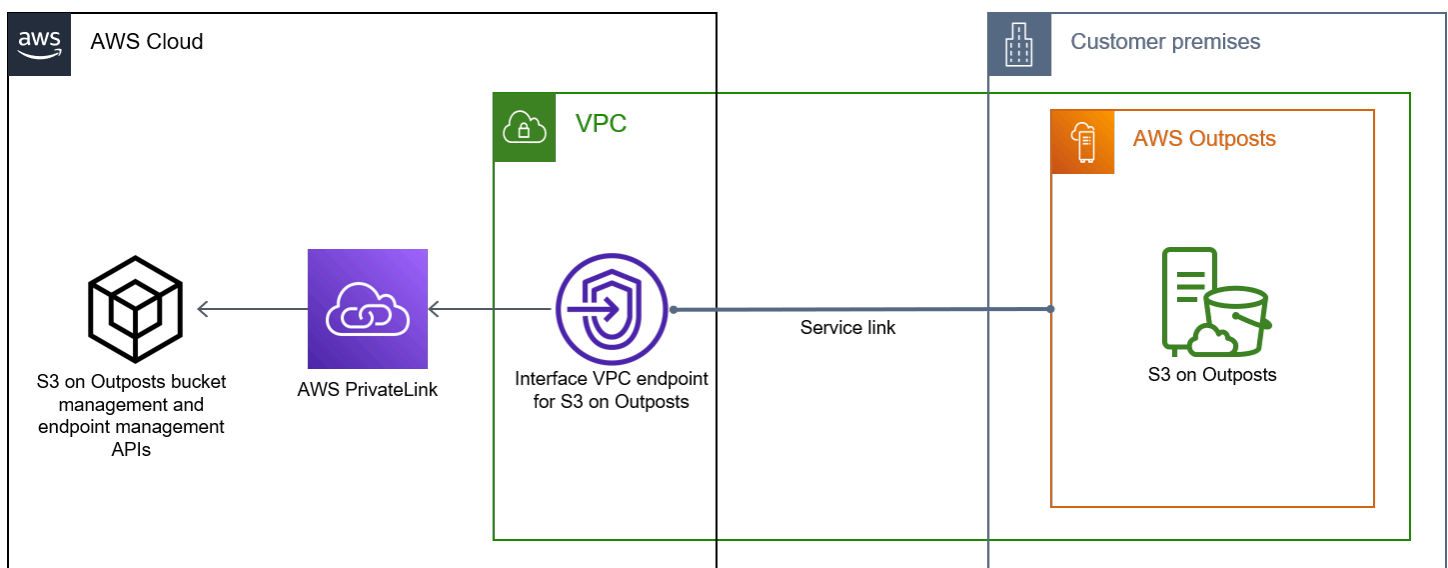
AWS PrivateLink für S3 auf Outposts

S3 on Outposts unterstützt AWS PrivateLink, was direkten Verwaltungszugriff auf Ihren S3 on Outposts-Speicher über einen privaten Endpunkt in Ihrem virtuellen privaten Netzwerk ermöglicht. Auf diese Weise können Sie Ihre interne Netzwerkarchitektur vereinfachen und Verwaltungsvorgänge auf Ihrem Outposts-Objektspeicher ausführen, indem Sie private IP-Adressen in Ihrer Virtual Private Cloud (VPC) verwenden. Durch die Verwendung AWS PrivateLink entfällt die Notwendigkeit, öffentliche IP-Adressen oder Proxyserver zu verwenden.

[Mit AWS PrivateLink for Amazon S3 on Outposts können Sie VPC-Schnittstellen-Endpunkte in Ihrer Virtual Private Cloud \(VPC\) bereitstellen, um auf Ihre S3 on Outposts Bucket Management- und Endpoint Management-APIs zuzugreifen.](#) Schnittstellen-VPC-Endpunkte sind direkt von Anwendungen aus zugänglich, die in Ihrer VPC oder On-Premises über Ihr Virtual Private Network (VPN) oder AWS Direct Connect bereitgestellt sind. Sie können auf die Bucket- und Endpoint Management-APIs zugreifen über AWS PrivateLink. AWS PrivateLink unterstützt keine API-Operationen [zur Datenübertragung](#) wie GET, PUT und ähnliche APIs. Diese Vorgänge werden bereits privat über die Konfiguration des S3-on-Outposts-Endpunkts und des Zugriffspunkts übertragen. Weitere Informationen finden Sie unter [Vernetzung für S3 on Outposts](#).

Schnittstellenendpunkte werden durch eine oder mehrere Elastic Network-Schnittstellen (ENIs) repräsentiert, denen private IP-Adressen aus Subnetzen in Ihrer VPC zugewiesen werden. Anfragen, die an Schnittstellenendpunkte für S3 on Outposts gestellt werden, werden automatisch an S3-on-Outposts-APIs zur Bucket- und Endpunktverwaltung im AWS-Netzwerk weitergeleitet. Sie können auch von lokalen Anwendungen aus über AWS Direct Connect oder AWS Virtual Private Network () auf Schnittstellenendpunkte in Ihrer VPC zugreifen. AWS VPN Weitere Informationen darüber, wie Sie Ihre VPC mit Ihrem On-Premises-Netzwerk verbinden, finden Sie im [AWS Direct Connect - Benutzerhandbuch](#) und im [AWS Site-to-Site VPN - Benutzerhandbuch](#).

Schnittstellenendpunkte leiten Anfragen für S3 auf Outposts-Bucket- und Endpoint Management-APIs über das AWS Netzwerk und durch das Netzwerk weiter AWS PrivateLink, wie in der folgenden Abbildung dargestellt.



Allgemeine Informationen zu Schnittstellen-Endpunkten finden Sie unter [VPC-Schnittstellen-Endpunkte \(AWS PrivateLink\)](#) im AWS PrivateLink -Handbuch.

Themen

- [Beschränkungen und Einschränkungen](#)
- [Zugriff auf S3-on-Outposts-Schnittstellenendpunkte](#)
- [Aktualisieren einer lokalen DNS-Konfiguration](#)
- [Erstellen eines VPC-Endpunkts für S3 on Outposts](#)
- [Erstellen von Bucket-Richtlinien und VPC-Endpunktrichtlinien für S3 on Outposts](#)

Beschränkungen und Einschränkungen

Wenn Sie über Outposts Bucket und Endpoint Management APIs auf S3 zugreifen AWS PrivateLink, gelten VPC-Einschränkungen. Weitere Informationen finden Sie unter [Interface endpoint properties and limitations \(Eigenschaften und Beschränkungen von Schnittstellen-Endpunkten\)](#) und [AWS PrivateLink quotas \(PrivateLink-Kontingente\)](#) im AWS PrivateLink -Leitfaden.

Darüber hinaus unterstützt AWS PrivateLink es Folgendes nicht:

- [Endpunkte für den Federal Information Processing Standard \(FIPS\)](#).
- [S3-on-Outposts-Datenübertragungs-APIs](#) z. B. GET, PUT und ähnliche Objekt-API-Operationen.
- Privates DNS

Zugriff auf S3-on-Outposts-Schnittstellenendpunkte

Um mithilfe von Outposts Bucket- und Endpoint Management-APIs auf S3 zuzugreifen AWS PrivateLink, müssen Sie Ihre Anwendungen so aktualisieren, dass sie endpunktspezifische DNS-Namen verwenden. Wenn Sie einen Schnittstellenendpunkt erstellen, werden zwei Typen von endpunktspezifischen S3-Namen für Outposts AWS PrivateLink generiert: Regional und Zonal.

- Regionale DNS-Namen — beinhalten eine eindeutige VPC-Endpunkt-ID, eine Service-ID, die AWS-Region, und `vpce.amazonaws.com`, zum Beispiel `vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com`
- Zonale DNS-Namen — beinhalten eine eindeutige VPC-Endpunkt-ID, die Availability Zone, eine Service-ID, die AWS-Region, und `vpce.amazonaws.com`, zum Beispiel `vpce-1a2b3c4d-5e6f-us-east-1a.s3-outposts.us-east-1.vpce.amazonaws.com`
Sie können diese Option verwenden, wenn Ihre Architektur Availability Zones isoliert. Sie könnten zonengebundene Namen beispielsweise zur Fehlereingrenzung oder zur Senkung der regionalen Datenübertragungskosten verwenden.

Important

Die Endpunkte der S3-on-Outposts-Schnittstelle werden von der öffentlichen DNS-Domain aus aufgelöst. S3 on Outposts unterstützt kein privates DNS. Benutze den Parameter `--endpoint-url` für alle Bucket- und Endpunktverwaltungs-APIs.

AWS CLI Beispiele

Verwenden Sie die Parameter `--region` und `--endpoint-url` für den Zugriff auf Bucket- und Endpunktverwaltungs-APIs über S3-on-Outposts-Schnittstellenendpunkte.

Example : Verwenden der Endpunkt-URL zum Auflisten von Buckets mit der S3-Steuerungs-API

Im folgenden Beispiel ersetzen Sie die Region `us-east-1`, die VPC-Endpunkt-URL `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com` und die Konto-ID `111122223333` durch entsprechende Informationen.

```
aws s3control list-regional-buckets --region us-east-1 --endpoint-url
https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com --account-
id 111122223333
```

AWS SDK-Beispiele

Aktualisieren Sie Ihre SDKs auf die neueste Version und konfigurieren Sie Ihre Clients so, dass sie eine Endpunkt-URL für den Zugriff auf eine S3-Steuerungs-API für S3-on-Outposts-Schnittstellenendpunkte verwenden. Weitere Informationen finden Sie unter [AWS -SDK-Beispiele für AWS PrivateLink](#).

SDK for Python (Boto3)

Example : Verwenden einer Endpunkt-URL, um auf die S3-Steuerungs-API zuzugreifen

Ersetzen Sie im folgenden Beispiel die Region `us-east-1` und die VPC-Endpunkt-URL `vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com` durch entsprechende Informationen.

```
control_client = session.client(
    service_name='s3control',
    region_name='us-east-1',
    endpoint_url='https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com'
)
```

Weitere Informationen finden Sie unter [AWS PrivateLink für Amazon S3](#) im Boto3-Entwicklerhandbuch.

SDK for Java 2.x

Example : Verwenden einer Endpunkt-URL, um auf die S3-Steuerungs-API zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC-Endpoint-URL *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* und die Region *Region.US_EAST_1* durch entsprechende Informationen.

```
// control client
Region region = Region.US_EAST_1;
S3ControlClient = S3ControlClient.builder().region(region)

    .endpointOverride(URI.create("https://vpce-1a2b3c4d-5e6f.s3-outposts.us-
east-1.vpce.amazonaws.com"))
        .build()
```

Weitere Informationen finden Sie unter [S3ControlClient](#) in der AWS SDK for Java -API-Referenz.

Aktualisieren einer lokalen DNS-Konfiguration

Wenn Sie endpunktspezifische DNS-Namen für den Zugriff auf die Schnittstellenendpunkte für S3-on-Outposts-APIs für die Bucket- und Endpunktverwaltung verwenden, brauchen Sie Ihren On-Premises-DNS-Resolver nicht zu aktualisieren. Sie können den endpunktspezifischen DNS-Namen mit der privaten IP-Adresse des Schnittstellenendpunkts aus der öffentlichen S3-on-Outposts-DNS-Domäne auflösen.

Erstellen eines VPC-Endpunkts für S3 on Outposts

Informationen zum Erstellen eines VPC-Schnittstellenendpunkts für S3 on Outposts finden Sie unter [Erstellen eines VPC-Endpunkts](#) im AWS PrivateLink -Handbuch.

Erstellen von Bucket-Richtlinien und VPC-Endpunktrichtlinien für S3 on Outposts

Sie können eine Endpunktrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff auf S3 on Outposts steuert. Sie können auch die `aws:sourceVpce`-Bedingung in S3-on-Outposts-Bucket-Richtlinien verwenden, um den Zugriff auf bestimmte Buckets von einem bestimmten VPC-Endpunkt aus zu beschränken. Mit VPC-Endpunktrichtlinien können Sie den Zugriff auf S3-on-Outposts-APIs für die Bucket- und Endpunktverwaltung steuern. Mit Bucket-Richtlinien können Sie den Zugriff

auf S3-on-Outposts-APIs für die Bucket-Verwaltung steuern. Sie können jedoch den Zugriff auf Objektaktionen für S3 on Outposts nicht mit `aws : sourceVpce` verwalten.

Zugriffsrichtlinien für S3 on Outposts enthalten die folgenden Informationen:

- Der AWS Identity and Access Management (IAM-) Principal, für den Aktionen zugelassen oder verweigert werden.
- Die S3-Steuerungsaktionen, die erlaubt oder verweigert werden.
- Die S3-on-Outposts-Ressourcen, für die Aktionen erlaubt oder verweigert werden.

Die folgenden Beispiele zeigen Richtlinien, die den Zugriff auf einen Bucket oder einen Endpunkt einschränken. Weitere Informationen zur VPC-Konnektivität finden Sie unter [Netzwerk-to-VPC-Konnektivitätsoptionen](#) im AWS Whitepaper [Amazon Virtual Private Cloud Connectivity Options](#).

Important

- Wenn Sie die in diesem Abschnitt beschriebenen Beispielrichtlinien für VPC-Endpunkte anwenden, können Sie Ihren Zugriff auf den Bucket unbeabsichtigt blockieren. Bucket-Berechtigungen, die den Bucket-Zugriff auf Verbindungen beschränken, die von Ihrem VPC-Endpunkt ausgehen, können alle Verbindungen mit dem Bucket blockieren. Informationen zur Behebung dieses Problems finden Sie unter [My bucket policy has the wrong VPC or VPC endpoint ID \(Meine Bucket-Richtlinie hat die falsche VPC- oder VPC-Endpunkt-ID\). Wie kann ich die Richtlinie so ändern, dass ich auf den Bucket zugreifen kann? im AWS Support Knowledge Center](#).
- Bevor Sie die folgende Bucket-Beispielrichtlinien verwenden, ersetzen Sie die VPC-Endpunkt-ID durch einen geeigneten Wert für Ihren Anwendungsfall. Andernfalls können Sie nicht auf Ihren Bucket zugreifen.
- Wenn Ihre Richtlinie nur den Zugriff auf einen S3-on-Outposts-Bucket von einem bestimmten VPC-Endpunkt aus erlaubt, deaktiviert sie den Konsolenzugriff für diesen Bucket, da die Konsolenanforderungen nicht vom angegebenen VPC-Endpunkt stammen.

Themen

- [Beispiel: Beschränken des Zugriffs auf einen bestimmten Bucket von einem VPC-Endpunkt aus](#)
- [Beispiel: Verweigern des Zugriffs von einem bestimmten VPC-Endpunkt aus in einer S3-on-Outposts-Bucket-Richtlinie](#)

Beispiel: Beschränken des Zugriffs auf einen bestimmten Bucket von einem VPC-Endpunkt aus

Sie können eine Endpunktrichtlinie erstellen, die den Zugriff auf bestimmte S3-on-Outposts-Buckets beschränkt. Die folgende Richtlinie beschränkt den Zugriff für die Aktion nur auf die `GetBucketPolicy` *example-outpost-bucket*. Zum Verwenden dieses Beispiels ersetzen Sie die Beispielwerte durch Ihre eigenen.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": {"AWS": "111122223333"},
      "Action": "s3-outposts:GetBucketPolicy",
      "Effect": "Allow",
      "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-
bucket"
    }
  ]
}
```

Beispiel: Verweigern des Zugriffs von einem bestimmten VPC-Endpunkt aus in einer S3-on-Outposts-Bucket-Richtlinie

Die folgende Bucket-Richtlinie für S3 on Outposts verweigert den Zugriff `GetBucketPolicy` auf den *example-outpost-bucket* Bucket über den *vpce-1a2b3c4d* VPC-Endpunkt.

Die `aws:sourceVpce`-Bedingung gibt den Endpunkt an und erfordert keinen Amazon-Ressourcennamen (ARN) für die VPC-Endpunkt-Ressource, sondern nur die Endpunkt-ID. Zum Verwenden dieses Beispiels ersetzen Sie die Beispielwerte durch Ihre eigenen.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Deny-access-to-specific-VPCE",
      "Principal": {"AWS": "111122223333"},
      "Action": "s3-outposts:GetBucketPolicy",
      "Effect": "Deny",

```

```

    "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-
bucket",
    "Condition": {
      "StringEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}
    }
  ]
}

```

AWS-Signature Version 4 (SigV4) – Authentifizierungsspezifische Richtlinienschlüssel

Die folgende Tabelle zeigt die Bedingungsschlüssel für die Authentifizierung mit AWS-Signature Version 4 (SigV4), die Sie mit Amazon S3 on Outposts verwenden können. In einer Bucket-Richtlinie können Sie diese Bedingungen hinzufügen, um ein bestimmtes Verhalten zu erzwingen, wenn Anforderungen mit der Signature Version 4 authentifiziert werden. Beispiele für Richtlinien finden Sie unter [Beispiele für Bucket-Richtlinien, die mit der Signature Version 4 verbundene Bedingungsschlüssel verwenden](#). Weitere Informationen zur Authentifizierung von Anfragen mit Signature Version 4 finden Sie unter [Authentifizieren von Anforderungen \(AWS-Signature Version 4\)](#) in der Amazon Simple Storage Service API-Referenz

Anwendbare Schlüssel für **s3-outposts:*** Aktionen oder eine der S3 on Outposts-Aktionen

Anwendbare Schlüssel	Beschreibung
s3-outposts:authType	<p>S3 on Outposts unterstützt verschiedene Methoden der Authentifizierung. Um eingehende Anfragen auf die Verwendung einer bestimmten Authentifizierungsmethode zu beschränken, können Sie diesen optionalen Bedingungsschlüssel verwenden. Sie können diesen Bedingungsschlüssel zum Beispiel verwenden, um nur den HTTP-Authzation -Header für die Authentifizierung von Anfragen zuzulassen.</p> <p>Zulässige Werte:</p> <p>REST-HEADER</p> <p>REST-QUERY-STRING</p>

Anwendbare Schlüssel	Beschreibung
s3-outposts:signatureAge	<p>Die Zeitspanne in Millisekunden, die eine Signatur in einer authentifizierten Anfrage gültig ist.</p> <p>Diese Bedingung gilt nur für vorseignierte URLs.</p> <p>In der Signature Version 4 ist der Signierschlüssel bis zu sieben Tage lang gültig. Daher sind die Signaturen auch bis zu sieben Tage lang gültig. Weitere Informationen finden Sie unter Einführung in das Signieren von Anfragen in der Amazon Simple Storage Service API-Referenz. Sie können diese Bedingung verwenden, um das Alter der Unterschrift weiter einzuschränken.</p> <p>Beispielwert: 600000</p>

Anwendbare Schlüssel	Beschreibung
s3-outposts:x-amz-content-sha256	<p>Sie können diesen Bedingungsschlüssel verwenden, um nicht signierte Inhalte in Ihrem Bucket zu verbieten.</p> <p>Wenn Sie die Signature Version 4 verwenden, fügen Sie bei Anfragen, die den <code>Authorization</code> Header verwenden, den <code>x-amz-content-sha256</code> Header in die Signaturberechnung ein und setzen dann seinen Wert auf die Hash-Nutzlast.</p> <p>Sie können diesen Bedingungsschlüssel in Ihrer Bucket-Richtlinie verwenden, um alle Uploads zu verweigern, deren Nutzdaten nicht signiert sind. Zum Beispiel:</p> <ul style="list-style-type: none"> • Verweigern Sie Uploads, die den <code>Authorization</code> -Header zur Authentifizierung von Anfragen verwenden, aber die Nutzdaten nicht signieren. Weitere Informationen finden Sie unter Übertragen von Nutzdaten in einem einzelnen Datenblock in der Amazon Simple Storage Service API-Referenz. • Verweigert Uploads, die vorsignierte URLs verwenden. Vorsignierte URLs haben immer eine <code>UNSIGNED_PAYLOAD</code> . Weitere Informationen finden Sie unter Authentifizierung von Anfragen und Authentifizierungsmethoden in der Amazon Simple Storage Service API-Referenz. <p>Zulässiger Wert: <code>UNSIGNED-PAYLOAD</code></p>

Beispiele für Bucket-Richtlinien, die mit der Signature Version 4 verbundene Bedingungsschlüssel verwenden

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Example : **s3-outposts:signatureAge**

Die folgende Bucket-Richtlinie verweigert jede S3 on Outposts vorsignierte URL-Anfrage auf Objekte in `example-outpost-bucket`, wenn die Signatur mehr als 10 Minuten alt ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Deny a presigned URL request if the signature is more than 10
minutes old",
      "Effect": "Deny",
      "Principal": {"AWS": "444455556666"},
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
      "Condition": {
        "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
        "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
      }
    }
  ]
}
```

Example : s3-outposts:authType

Die folgende Bucket-Richtlinie lässt nur Anfragen zu, die den Authorization-Header für die Anfrageauthentifizierung verwenden. Alle vorsignierten URL-Anfragen werden abgelehnt, da vorsignierte URLs Abfrageparameter verwenden, um Anfrage- und Authentifizierungsinformationen bereitzustellen. Weitere Informationen finden Sie unter [Authentifizierungsmethoden](#) in der Amazon Simple Storage Service API-Referenz.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow only requests that use the Authorization header for
request authentication. Deny presigned URL requests.",
      "Effect": "Deny",
      "Principal": {"AWS": "111122223333"},
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
      "Condition": {
        "StringNotEquals": {
```

```

        "s3-outposts:authType": "REST-HEADER"
      }
    }
  ]
}

```

Example : **s3-outposts:x-amz-content-sha256**

Die folgende Bucket-Richtlinie verweigert alle Uploads mit unsignierten Nutzdaten, z. B. Uploads, die vorsignierte URLs verwenden. Weitere Informationen finden Sie unter [Authentifizierung von Anfragen](#) und [Authentifizierungsmethoden](#) in der Amazon Simple Storage Service API-Referenz.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Deny uploads with unsigned payloads.",
      "Effect": "Deny",
      "Principal": {"AWS": "111122223333"},
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/*",
      "Condition": {
        "StringEquals": {
          "s3-outposts:x-amz-content-sha256": "UNSIGNED-PAYLOAD"
        }
      }
    }
  ]
}

```

AWS-verwaltete Richtlinien für Amazon S3 on Outposts

Eine von AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Von AWS verwaltete Richtlinien stellen Berechtigungen für viele häufige Anwendungsfälle bereit, damit Sie beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise nicht die geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind.

Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [kundenverwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Die Berechtigungen, die in den von AWS verwalteten Richtlinien definiert sind, können nicht geändert werden. Wenn AWS Berechtigungen aktualisiert, die in einer von AWS verwalteten Richtlinie definiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am wahrscheinlichsten eine von AWS verwaltete Richtlinie, wenn ein neuer AWS-Service gestartet wird oder neue API-Operationen für bestehende Services verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

AWS-verwaltete Richtlinie: AWSS3OnOutPostsServiceRolePolicy

Hilft Ihnen im Rahmen der serviceverknüpften Rolle `AWSServiceRoleForS3OnOutposts` bei der Verwaltung von Netzwerkressourcen.

Informationen zum Anzeigen der Berechtigungen für diese Richtlinie finden Sie unter [AWSS3OnOutpostsServiceRolePolicy](#).

Änderungen von S3 on Outposts zu AWS-verwalteten Richtlinien

Sehen Sie sich Details über Aktualisierungen an von AWS verwalteten Richtlinien für S3 on Outposts an, seit der Service diese Änderungen nachverfolgt.

Änderung	Beschreibung	Datum
S3 on Outposts hat <code>AWSS3OnOutpostsServiceRolePolicy</code> hinzugefügt	S3 on Outposts hat <code>AWSS3OnOutpostsServiceRolePolicy</code> als Teil der serviceverknüpften Rolle <code>AWSServiceRoleForS3OnOutposts</code> hinzugefügt, die bei der Verwaltung von Netzwerkressourcen hilft.	03. Oktober 2023
S3 on Outposts hat mit der Verfolgung von Änderungen begonnen	S3 on Outposts hat mit der Verfolgung von Änderungen	03. Oktober 2023

Änderung	Beschreibung	Datum
	n für seine AWS-verwaltete Richtlinien begonnen.	

Verwenden von serviceverknüpften Rollen für S3 on Outposts

Amazon S3 on Outposts verwendet mit AWS Identity and Access Management (IAM) [verknüpfte Servicerollen](#). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit S3 on Outposts verknüpft ist. Serviceverknüpfte Rollen werden von S3 on Outposts vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht das Einrichten von S3 on Outposts, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. S3 on Outposts definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur S3 on Outposts die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre S3-on-Outposts-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rollen angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Berechtigungen von serviceverknüpften Rollen für S3 on Outposts

S3 on Outposts verwendet die serviceverknüpfte Rolle `AWSServiceRoleForS3OnOutposts`, um Sie bei der Verwaltung von Netzwerkressourcen zu unterstützen.

Die serviceverknüpfte Rolle `AWSServiceRoleForS3OnOutposts` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `s3-outposts.amazonaws.com`

Die Rollenberechtigungsrichtlinie `AWSS3OnOutpostsServiceRolePolicy` ermöglicht S3 on Outposts die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeVpcs",
      "ec2:DescribeCoipPools",
      "ec2:GetCoipPoolUsage",
      "ec2:DescribeAddresses",
      "ec2:DescribeLocalGatewayRouteTableVpcAssociations"
    ],
    "Resource": "*",
    "Sid": "DescribeVpcResources"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterface"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:subnet/*",
      "arn:aws:ec2:*:*:security-group/*"
    ],
    "Sid": "CreateNetworkInterface"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterface"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:network-interface/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/CreatedBy": "S3 On Outposts"
      }
    }
  }
}
```

```

    },
    "Sid": "CreateTagsForCreateNetworkInterface"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AllocateAddress"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:ipv4pool-ec2/*"
    ],
    "Sid": "AllocateIpAddress"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AllocateAddress"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:elastic-ip/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/CreatedBy": "S3 On Outposts"
      }
    },
    "Sid": "CreateTagsForAllocateIpAddress"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:ModifyNetworkInterfaceAttribute",
      "ec2:CreateNetworkInterfacePermission",
      "ec2>DeleteNetworkInterface",
      "ec2>DeleteNetworkInterfacePermission",
      "ec2:DisassociateAddress",
      "ec2:ReleaseAddress",
      "ec2:AssociateAddress"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/CreatedBy": "S3 On Outposts"
      }
    }
  }

```



```
    },
    "Sid": "ReleaseVpcResources"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": [
          "CreateNetworkInterface",
          "AllocateAddress"
        ],
        "aws:RequestTag/CreatedBy": [
          "S3 On Outposts"
        ]
      }
    },
    "Sid": "CreateTags"
  }
]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. eine Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für S3 on Outposts

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen Endpunkt für S3 on Outposts in der AWS Management Console-, der AWS CLI- oder der AWS-API erstellen, erstellt S3 on Outposts die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Endpunkt für S3 on Outposts erstellen, erstellt S3 on Outposts die serviceverknüpfte Rolle erneut für Sie.

Sie können auch die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit dem Anwendungsfall S3 on Outposts zu erstellen. Erstellen Sie in der AWS CLI oder der AWS-API

eine servicegebundene Rolle mit dem Servicenamen `s3-outposts.amazonaws.com`. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch. Wenn Sie diese servicegebundene Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer serviceverknüpften Rolle für S3 on Outposts

S3 on Outposts verhindert die Bearbeitung der `AWSServiceRoleForS3OnOutposts` serviceverknüpften Rolle. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach der Erstellung nicht berücksichtigt werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für S3 on Outposts

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der S3-on-Outposts-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie Ressourcen von S3 on Outposts, die von der Rolle „`AWSServiceRoleForS3OnOutposts`“ verwendet werden

1. [Löschen Sie die S3-on-Outposts-Endpunkte](#) in Ihrem AWS-Konto in allen AWS-Regionen.
2. Löschen Sie die serviceverknüpfte Rolle mit IAM.

Verwenden Sie die IAM-Konsole, AWS CLI- oder AWS-API, um die `AWSServiceRoleForS3OnOutposts` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte S3-on-Outposts-Rollen

S3 on Outposts unterstützt die Verwendung von serviceverknüpften Rollen in allen AWS-Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [S3-on-Outposts-Regionen und -Endpunkte](#).

Verwaltung von S3-on-Outposts-Speicher

Mit Amazon S3 in Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Weitere Informationen zum Verwalten und Freigeben Ihrer Speicherkapazität von Amazon S3 in Outposts finden Sie in den folgenden Themen.

Themen

- [Verwalten der S3-Versionierung für Ihren S3-on-Outposts-Bucket](#)
- [Erstellen und Verwalten einer Lebenszyklus-Konfiguration für Ihren Amazon-S3-on-Outposts-Bucket](#)
- [Replikation von Objekten für S3 in Outposts](#)
- [S3 auf Outposts teilen mit AWS RAM](#)
- [Sonstige AWS-Services, die S3 on Outposts verwenden](#)

Verwalten der S3-Versionierung für Ihren S3-on-Outposts-Bucket

Wenn diese Option aktiviert ist, speichert die S3-Versionsverwaltung mehrere unterschiedliche Kopien eines Objekts im selben Bucket. Sie können die S3-Versionsverwaltung verwenden, um sämtliche Versionen aller Objekte in Ihren Outposts-Buckets zu speichern, abzurufen

oder wiederherzustellen. Mit der S3-Versionsverwaltung können Sie Versionen sowohl nach unbeabsichtigten Benutzeraktionen als auch nach Anwendungsfehlern problemlos wiederherstellen.

Buckets von Amazon S3 on Outposts verfügen über drei Versionsverwaltungsstatus:

- **Unversioned (Nicht versioniert)** – Wenn Sie die S3-Versionsverwaltung für Ihren Bucket noch nie aktiviert oder ausgesetzt haben, ist er nicht versioniert und gibt keinen S3-Versionsverwaltungsstatus zurück. Weitere Informationen über das S3-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).
- **Enabled (Aktiviert)** – Aktiviert die S3-Versionsverwaltung für die Objekte im Bucket. Alle Objekte, die dem Bucket hinzugefügt werden, erhalten eine eindeutige Versions-ID. Objekte, die zum Zeitpunkt der Aktivierung des Versioning bereits im Bucket vorhanden waren, haben die Versions-ID null. Wenn Sie diese (oder andere) Objekte mit anderen Operationen wie [PutObject](#) verändern, erhalten die neuen Objekte eine eindeutige Versions-ID.
- **Suspended (Ausgesetzt)** – Setzt die S3-Versionsverwaltung für die Objekte im Bucket aus. Alle Objekte, die dem Bucket hinzugefügt werden, nachdem die Versionsverwaltung ausgesetzt wurde, erhalten die Versions-ID null. Weitere Informationen finden Sie unter [Hinzufügen von Objekten zu Buckets mit ausgesetztem Versioning](#).

Nachdem Sie die S3-Versionsverwaltung für einen S3-on-Outposts-Bucket aktiviert haben, kann er nicht mehr auf einen nicht versionierten Status zurückgesetzt werden. Sie können die Versionsverwaltung jedoch aussetzen. Weitere Informationen über das S3-Versioning finden Sie unter [Verwenden der Versioning in S3-Buckets](#).

Sie haben für jedes Objekt in Ihrem Bucket eine aktuelle Version und keine oder mehr vorherige Versionen. Damit die Speicherkosten gesenkt werden, können Sie die S3-Lebenszyklusregeln für Ihren Bucket so konfigurieren, dass vorherige Versionen nach einem bestimmten Zeitraum ablaufen. Weitere Informationen finden Sie unter [Erstellen und Verwalten einer Lebenszyklus-Konfiguration für Ihren Amazon-S3-on-Outposts-Bucket](#).

Die folgenden Beispiele veranschaulichen, wie Sie die Versionsverwaltung für einen vorhandenen S3-on-Outposts-Bucket mithilfe der AWS Management Console und der AWS Command Line Interface (AWS CLI) aktivieren oder aussetzen. Informationen zum Erstellen eines Buckets mit aktivierter S3-Versionsverwaltung finden Sie unter [Erstellen eines S3-on-Outposts-Buckets](#).

Note

Das AWS-Konto, das den Bucket erstellt, besitzt ihn und ist das einzige, das ihm Aktionen zuweisen kann. Buckets verfügen über Konfigurationseigenschaften wie Outpost, Tags, Standard-Verschlüsselung und Zugriffspunkteinstellungen. Zu den Zugriffspunkteinstellungen gehören die Virtual Private Cloud (VPC), die Zugriffspunkt-Richtlinie für den Zugriff auf die Objekte im Bucket sowie andere Metadaten. Weitere Informationen finden Sie unter [Spezifikationen für S3 auf Outposts](#).

Verwenden der S3-Konsole

So bearbeiten Sie die S3-Versionsverwaltungseinstellungen für Ihren Bucket

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, für den Sie die S3-Versionsverwaltung aktivieren möchten.
4. Wählen Sie die Registerkarte Properties (Eigenschaften) aus.
5. Wählen Sie unter Bucket Versioning (Bucket-Versioning) die Option Edit (Bearbeiten).
6. Bearbeiten Sie die S3-Versionsverwaltungseinstellung für den Bucket, indem Sie eine der folgenden Optionen auswählen:
 - Wenn Sie die S3-Versionsverwaltung aussetzen und die Erstellung neuer Objektversionen anhalten möchten, wählen Sie Suspend (Aussetzen) aus.
 - Möchten Sie die S3-Versionsverwaltung aktivieren und mehrere unterschiedliche Kopien jedes Objekts speichern, wählen Sie Enable (Aktivieren) aus.
7. Wählen Sie Save Changes (Änderungen speichern).

Verwendung der AWS CLI

Wenn Sie die S3-Versionsverwaltung für Ihren Bucket mithilfe der AWS CLI aktivieren oder aussetzen möchten, verwenden Sie den Befehl `put-bucket-versioning`, wie in den folgenden Beispielen gezeigt. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

Weitere Informationen finden Sie unter [put-bucket-versioning](#) in der AWS CLI-Referenz.

Example : S3-Versionsverwaltung aktivieren

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --versioning-configuration Status=Enabled
```

Example : S3-Versionsverwaltung aussetzen

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --versioning-configuration Status=Suspended
```

Erstellen und Verwalten einer Lebenszyklus-Konfiguration für Ihren Amazon-S3-on-Outposts-Bucket

Sie können S3 Lifecycle verwenden, um die Speicherkapazität für Amazon S3 on Outposts zu optimieren. Sie können Lebenszyklusregeln erstellen, um Objekte ablaufen zu lassen, wenn sie veralten oder durch neuere Versionen ersetzt werden. Sie können eine Lebenszyklusregel erstellen, aktivieren, deaktivieren oder löschen.

Weitere Informationen zum S3-Lebenszyklus finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Note

Das AWS-Konto, das den Bucket erstellt, besitzt ihn und ist das einzige, das eine Lebenszyklusregel erstellen, aktivieren, deaktivieren oder löschen kann.

Informationen zum Erstellen und Verwalten der Lebenszyklus-Konfiguration für Ihren S3-on-Outposts-Bucket finden Sie in den folgenden Themen.

Themen

- [Erstellen und Verwalten einer Lebenszyklusregel mithilfe der AWS Management Console](#)
- [Erstellen und Verwalten einer Lebenszyklus-Konfiguration mithilfe der AWS CLI und dem SDK for Java](#)

Erstellen und Verwalten einer Lebenszyklusregel mithilfe der AWS Management Console

Sie können S3 Lifecycle verwenden, um die Speicherkapazität für Amazon S3 on Outposts zu optimieren. Sie können Lebenszyklusregeln erstellen, um Objekte ablaufen zu lassen, wenn sie veralten oder durch neuere Versionen ersetzt werden. Sie können eine Lebenszyklusregel erstellen, aktivieren, deaktivieren oder löschen.

Weitere Informationen zum S3-Lebenszyklus finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Note

Das AWS-Konto, das den Bucket erstellt, besitzt ihn und ist das einzige, das eine Lebenszyklusregel erstellen, aktivieren, deaktivieren oder löschen kann.

Weitere Informationen zum Erstellen und Verwalten einer Lebenszyklusregel für S3 on Outposts mithilfe der AWS Management Console finden Sie in den folgenden Themen.

Themen


- [Erstellen einer Lebenszyklusregel](#)
- [Aktivieren einer Lebenszyklusregel](#)
- [Bearbeiten einer Lebenszyklusregel](#)
- [Löschen einer Lebenszyklusregel](#)

Erstellen einer Lebenszyklusregel

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, für den Sie eine Lebenszyklusregel erstellen möchten.
4. Wählen Sie die Registerkarte Management (Verwaltung) und dann die Option Create lifecycle rule (Lebenszyklusregel erstellen) aus.
5. Geben Sie einen Wert für Lifecycle rule name (Lebenszyklusregelname) ein.
6. Wählen Sie unter Rule scope (Regelumfang) eine der folgenden Optionen aus:

- Wenn Sie den Umfang mit bestimmten Filtern einschränken möchten, wählen Sie **Limit the scope of this rule using one or more filters** (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken) aus. Fügen Sie anschließend einen Präfixfilter, Tags oder eine Objektgröße hinzu.
 - Wenn Sie diese Lebenszyklusregel auf alle Objekte im Bucket anwenden möchten, wählen Sie **Apply to all objects in the bucket** (Auf alle Objekte im Bucket anwenden) aus.
7. Wählen Sie unter **Lifecycle rule actions** (Lebenszyklusregelaktionen) eine der folgenden Optionen aus:
- **Expire current versions of objects** (Aktuelle Objektversionen ablaufen lassen) – Bei Buckets mit aktivierter Versionsverwaltung fügt S3 on Outposts eine Löschmarkierung hinzu und behält die Objekte als nicht aktuelle Versionen bei. Bei Buckets, die keine S3-Versionsverwaltung verwenden, löscht S3 on Outposts die Objekte dauerhaft.
 - **Permanently delete noncurrent versions of objects** (Vorherige Objektversionen dauerhaft löschen) – S3 on Outposts löscht nicht aktuelle Objektversionen dauerhaft.
 - **Delete expired object delete markers or incomplete multipart uploads** (Abgelaufene Objektlöschmarkierungen oder unvollständige mehrteilige Uploads löschen) – S3 on Outposts löscht Löschmarkierungen für abgelaufene Objekte oder unvollständige mehrteilige Uploads dauerhaft.

Wenn Sie den Umfang Ihrer Lebenszyklusregel mithilfe von Objekt-Tags einschränken, können Sie die Option **Delete expired object delete markers** (Löschmarkierungen für abgelaufenes Objekt löschen) nicht auswählen. Sie können die Option **Delete expired object delete markers** (Löschmarkierungen für abgelaufenes Objekt löschen) auch nicht auswählen, wenn Sie **Expire current object versions** (Aktuelle Objektversionen ablaufen lassen) aktiviert haben.

 **Note**

Größenabhängige Filter können nicht mit Löschmarkierungen und unvollständigen mehrteiligen Uploads verwendet werden.

8. Wenn Sie **Expire current versions of objects** (Aktuelle Objektversionen ablaufen lassen) oder **Permanently delete noncurrent versions of objects** (Vorherige Objektversionen dauerhaft

löschen) ausgewählt haben, konfigurieren Sie den Regelauslöser basierend auf einem bestimmten Datum oder dem Alter des Objekts.

9. Wenn Sie die Option Delete expired object delete markers (Löschmarkierungen für abgelaufenes Objekt löschen) ausgewählt haben, wählen Sie zur Bestätigung dieses Vorgangs die Option Delete expired object delete markers (Löschmarkierungen für abgelaufenes Objekt löschen) erneut aus.
10. Überprüfen Sie unter Timeline Summary (Timeline-Zusammenfassung) Ihre Lebenszyklusregel und wählen Sie Create rule (Regel erstellen) aus.

Aktivieren einer Lebenszyklusregel

So aktivieren oder deaktivieren Sie eine Bucket-Lebenszyklusregel

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, für den Sie eine Lebenszyklusregel deaktivieren möchten.
4. Wählen Sie die Registerkarte Management (Verwaltung) und dann unter Lifecycle rule (Lebenszyklusregel) die Regel aus, die Sie aktivieren oder deaktivieren möchten.
5. Wählen Sie für Aktion die Option Regel aktivieren oder deaktivieren aus.

Bearbeiten einer Lebenszyklusregel

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, für den Sie eine Lebenszyklusregel bearbeiten möchten.
4. Wählen Sie die Registerkarte Verwaltung und wählen Sie die Lebenszyklusregel aus, die Sie bearbeiten möchten.
5. (Optional) Aktualisieren Sie den Wert für Lifecycle rule name (Lebenszyklusregelname).
6. Bearbeiten Sie unter Rule scope (Regelumfang) den Umfang nach Bedarf:
 - Wenn Sie den Umfang mit bestimmten Filtern einschränken möchten, wählen Sie Limit the scope of this rule using one or more filters (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken) aus. Fügen Sie anschließend einen Präfixfilter, Tags oder eine Objektgröße hinzu.

- Wenn Sie diese Lebenszyklusregel auf alle Objekte im Bucket anwenden möchten, wählen Sie **Apply to all objects in the bucket** (Auf alle Objekte im Bucket anwenden) aus.
7. Wählen Sie unter **Lifecycle rule actions** (Lebenszyklusregelaktionen) eine der folgenden Optionen aus:
- **Expire current versions of objects** (Aktuelle Objektversionen ablaufen lassen) – Bei Buckets mit aktivierter Versionsverwaltung fügt S3 on Outposts eine Löschmarkierung hinzu und behält die Objekte als nicht aktuelle Versionen bei. Bei Buckets, die keine S3-Versionsverwaltung verwenden, löscht S3 on Outposts die Objekte dauerhaft.
 - **Permanently delete noncurrent versions of objects** (Vorherige Objektversionen dauerhaft löschen) – S3 on Outposts löscht nicht aktuelle Objektversionen dauerhaft.
 - **Delete expired object delete markers or incomplete multipart uploads** (Abgelaufene Objektlöschmarkierungen oder unvollständige mehrteilige Uploads löschen) – S3 on Outposts löscht Löschmarkierungen für abgelaufene Objekte oder unvollständige mehrteilige Uploads dauerhaft.

Wenn Sie den Umfang Ihrer Lebenszyklusregel mithilfe von Objekt-Tags einschränken, können Sie die Option **Delete expired object delete markers** (Löschmarkierungen für abgelaufenes Objekt löschen) nicht auswählen. Sie können die Option **Delete expired object delete markers** (Löschmarkierungen für abgelaufenes Objekt löschen) auch nicht auswählen, wenn Sie **Expire current object versions** (Aktuelle Objektversionen ablaufen lassen) aktiviert haben.

 Note

Größenabhängige Filter können nicht mit Löschmarkierungen und unvollständigen mehrteiligen Uploads verwendet werden.

8. Wenn Sie **Expire current versions of objects** (Aktuelle Objektversionen ablaufen lassen) oder **Permanently delete noncurrent versions of objects** (Vorherige Objektversionen dauerhaft löschen) ausgewählt haben, konfigurieren Sie den Regelauslöser basierend auf einem bestimmten Datum oder dem Objektalter.
9. Wenn Sie die Option **Delete expired object delete markers** (Löschmarkierungen für abgelaufenes Objekt löschen) ausgewählt haben, wählen Sie zur Bestätigung dieses Vorgangs die Option **Delete expired object delete markers** (Löschmarkierungen für abgelaufenes Objekt löschen) erneut aus.

10. Wählen Sie Save (Speichern).

Löschen einer Lebenszyklusregel

1. Öffnen Sie die Amazon S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Outposts-Bucket aus, für den Sie eine Lebenszyklusregel löschen möchten.
4. Wählen Sie die Registerkarte Management (Verwaltung) und unter Lifecycle rule (Lebenszyklusregel) die Regel aus, die Sie löschen möchten.
5. Wählen Sie Delete (Löschen).

Erstellen und Verwalten einer Lebenszyklus-Konfiguration mithilfe der AWS CLI und dem SDK for Java

Sie können S3 Lifecycle verwenden, um die Speicherkapazität für Amazon S3 on Outposts zu optimieren. Sie können Lebenszyklusregeln erstellen, um Objekte ablaufen zu lassen, wenn sie veralten oder durch neuere Versionen ersetzt werden. Sie können eine Lebenszyklusregel erstellen, aktivieren, deaktivieren oder löschen.

Weitere Informationen zum S3-Lebenszyklus finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Note

Das AWS-Konto, das den Bucket erstellt, besitzt ihn und ist das einzige, das eine Lebenszyklusregel erstellen, aktivieren, deaktivieren oder löschen kann.

Weitere Informationen zum Erstellen und Verwalten einer Lebenszyklus-Konfiguration für einen S3-on-Outposts-Bucket mithilfe der AWS Command Line Interface (AWS CLI) und von AWS SDK for Java finden Sie in den folgenden Beispielen.

Themen

- [PUT-Befehl für eine Lebenszyklus-Konfiguration](#)
- [GET-Befehl für eine Lebenszyklus-Konfiguration für einen S3-on-Outposts-Bucket](#)

PUT-Befehl für eine Lebenszyklus-Konfiguration

AWS CLI

Im folgenden AWS CLI-Beispiel wird eine Lebenszyklus-Konfigurationsrichtlinie in einen Outposts-Bucket eingefügt. Diese Richtlinie gibt an, dass alle Objekte mit dem gekennzeichneten Präfix (*myprefix*) und Tags nach 10 Tagen ablaufen. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

1. Speichern Sie die Richtlinie für die Lebenszyklus-Konfiguration in einer JSON-Datei. In diesem Beispiel heißt die Datei `lifecycle1.json`.

```
{
  "Rules": [
    {
      "ID": "id-1",
      "Filter": {
        "And": {
          "Prefix": "myprefix",
          "Tags": [
            {
              "Value": "mytagvalue1",
              "Key": "mytagkey1"
            },
            {
              "Value": "mytagvalue2",
              "Key": "mytagkey2"
            }
          ],
          "ObjectSizeGreaterThan": 1000,
          "ObjectSizeLessThan": 5000
        }
      },
      "Status": "Enabled",
      "Expiration": {
        "Days": 10
      }
    }
  ]
}
```

2. Senden Sie die JSON-Datei als Teil des CLI-Befehls `put-bucket-lifecycle-configuration`. Zum Verwenden dieses Befehls ersetzen Sie *user input*

placeholder durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [put-bucket-lifecycle-configuration](#) in der AWS CLI-Referenz.

```
aws s3control put-bucket-lifecycle-configuration --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket --lifecycle-configuration file://lifecycle1.json
```

SDK for Java

Im folgenden SDK-für-Java-Beispiel wird eine Lebenszyklus-Konfiguration in einen Outposts-Bucket eingefügt. Diese Lebenszykluskonfiguration gibt an, dass alle Objekte mit dem gekennzeichneten Präfix (*myprefix*) und Tags nach 10 Tagen ablaufen. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen. Weitere Informationen finden Sie unter [PutBucketLifecycleConfiguration](#) in der API-Referenz zu Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;

public void putBucketLifecycleConfiguration(String bucketArn) {

    S3Tag tag1 = new S3Tag().withKey("mytagkey1").withValue("mytagkey1");
    S3Tag tag2 = new S3Tag().withKey("mytagkey2").withValue("mytagkey2");

    LifecycleRuleFilter lifecycleRuleFilter = new LifecycleRuleFilter()
        .withAnd(new LifecycleRuleAndOperator()
            .withPrefix("myprefix")
            .withTags(tag1, tag2))
            .withObjectSizeGreaterThan(1000)
            .withObjectSizeLessThan(5000);

    LifecycleExpiration lifecycleExpiration = new LifecycleExpiration()
        .withExpiredObjectDeleteMarker(false)
        .withDays(10);

    LifecycleRule lifecycleRule = new LifecycleRule()
        .withStatus("Enabled")
        .withFilter(lifecycleRuleFilter)
        .withExpiration(lifecycleExpiration)
        .withID("id-1");

    LifecycleConfiguration lifecycleConfiguration = new LifecycleConfiguration()
```

```
        .withRules(lifecycleRule);

    PutBucketLifecycleConfigurationRequest reqPutBucketLifecycle = new
    PutBucketLifecycleConfigurationRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withLifecycleConfiguration(lifecycleConfiguration);

    PutBucketLifecycleConfigurationResult respPutBucketLifecycle =
    s3ControlClient.putBucketLifecycleConfiguration(reqPutBucketLifecycle);
    System.out.printf("PutBucketLifecycleConfiguration Response: %s%n",
    respPutBucketLifecycle.toString());
}
```

GET-Befehl für eine Lebenszyklus-Konfiguration für einen S3-on-Outposts-Bucket

AWS CLI

Im folgenden Beispiel AWS CLI wird eine Lebenszykluskonfiguration für einen Outposts-Bucket abgerufen. Zum Verwenden dieses Befehls ersetzen Sie *user input placeholder* durch Ihre eigenen Informationen. Weitere Informationen über diesen Befehl finden Sie unter [get-bucket-lifecycle-configuration](#) in der AWS CLI-Referenz.

```
aws s3control get-bucket-lifecycle-configuration --account-id 123456789012 --bucket
arn:aws:s3-outposts:<your-region>:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket
```

SDK for Java

Das folgende SDK für Java-Beispiel ruft eine Lebenszykluskonfiguration für einen Outposts-Bucket ab. Weitere Informationen finden Sie unter [GetBucketLifecycleConfiguration](#) in der API-Referenz zu Amazon Simple Storage Service.

```
import com.amazonaws.services.s3control.model.*;

public void getBucketLifecycleConfiguration(String bucketArn) {

    GetBucketLifecycleConfigurationRequest reqGetBucketLifecycle = new
    GetBucketLifecycleConfigurationRequest()
        .withAccountId(AccountId)
```

```
        .withBucket(bucketArn);

        GetBucketLifecycleConfigurationResult respGetBucketLifecycle =
s3ControlClient.getBucketLifecycleConfiguration(reqGetBucketLifecycle);
        System.out.printf("GetBucketLifecycleConfiguration Response: %s%n",
respGetBucketLifecycle.toString());
    }
```

Replikation von Objekten für S3 in Outposts

Mit S3 Replication in AWS Outposts können Sie Amazon S3 in Outposts so konfigurieren, dass S3-Objekte automatisch über verschiedene Outposts hinweg oder zwischen Buckets in demselben Outpost repliziert werden. Sie können S3 Replication in Outposts verwenden, um mehrere Replikate Ihrer Daten in denselben oder verschiedenen Outposts oder über verschiedene Konten hinweg zu verwalten und die Anforderungen an die Datenspeicherorte zu erfüllen. S3 Replication in Outposts hilft Ihnen dabei, Ihre Anforderungen an konforme Speicher und die Datenfreigabe zwischen verschiedenen Konten zu erfüllen. Wenn Sie sicherstellen müssen, dass Ihre Replikate mit den Quelldaten übereinstimmen, können Sie mit S3 Replication in Outposts Replikate Ihrer Objekte erstellen, die alle Metadaten enthalten, z. B. die Erstellungszeit des ursprünglichen Objekts, Tags und Versions-IDs.

S3 Replication in Outposts stellt außerdem detaillierte Metriken und Benachrichtigungen zum Überwachen des Status der Objektreplication zwischen Buckets bereit. Sie können Amazon CloudWatch verwenden, um den Replikationsfortschritt zu überwachen, indem Sie die Bytes mit ausstehender Replikation, die Operationen mit ausstehender Replikation und die Replikationslatenz zwischen Ihren Quell- und Ziel-Buckets nachverfolgen. Um Konfigurationsprobleme schnell zu diagnostizieren und zu beheben, können Sie außerdem Amazon EventBridge so einrichten, dass Benachrichtigungen über Fehler bei Replikationsobjekten empfangen werden. Weitere Informationen hierzu finden Sie unter [Verwalten Ihrer Replikation](#).

Themen

- [Replikations-Konfiguration](#)
- [Anforderungen für S3 Replication in Outposts](#)
- [Was wird repliziert?](#)
- [Was wird nicht repliziert?](#)
- [Was wird von S3 Replication in Outposts nicht unterstützt?](#)

- [Einrichten der Replikation](#)
- [Verwalten Ihrer Replikation](#)

Replikations-Konfiguration

S3 in Outposts speichert Replikations-Konfigurationen als XML. In der XML-Datei mit der Replikations-Konfiguration legen Sie eine AWS Identity and Access Management (IAM)-Rolle und mindestens eine Regel fest.

```
<ReplicationConfiguration>
  <Role>IAM-role-ARN</Role>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
  ...
</ReplicationConfiguration>
```

S3 in Outposts kann Objekte nur dann replizieren, wenn Sie die entsprechende Berechtigung erteilen. Sie erteilen S3 in Outposts Berechtigungen mit der IAM-Rolle, die Sie in der Replikations-Konfiguration angeben. S3 in Outposts übernimmt diese IAM-Rolle, um Objekte in Ihrem Namen zu replizieren. Sie müssen der IAM-Rolle die erforderlichen Berechtigungen erteilen, bevor Sie die Replikation starten. Weitere Informationen zu diesen Berechtigungen für S3 in Outposts finden Sie unter [Erstellen einer IAM-Rolle](#).

In den folgenden Szenarien fügen Sie eine Regel zur Replikationskonfiguration hinzu:

- Sie möchten alle Objekte replizieren.
- Sie möchten eine Teilmenge der Objekte replizieren. Sie identifizieren die Teilmenge der Objekte, indem Sie einen Filter zur Regel hinzufügen. In dem Filter geben Sie ein Objektschlüsselpräfix, Markierungen oder eine Kombination aus beidem an, um die Objektteilmenge zu identifizieren, für die die Regel gilt.

Sie fügen mehrere Regeln zu einer Replikationskonfiguration hinzu, wenn Sie eine andere Teilmenge von Objekten replizieren möchten. In jeder Regel geben Sie einen Filter an, der eine andere Teilmenge von Objekten auswählt. Beispiel: Sie möchten Objekte mit dem Schlüsselpräfix `tax/` oder

document/ replizieren. Dazu fügen Sie zwei Regeln hinzu, eine, die den tax/-Schlüsselpräfix-Filter angibt und eine andere, die das document/-Schlüsselpräfix angibt.

Weitere Informationen zur Replikations-Konfiguration und zu den Replikationsregeln von S3 in Outposts finden Sie unter [ReplicationConfiguration](#) in der API-Referenz zu Amazon Simple Storage Service.

Anforderungen für S3 Replication in Outposts

Für die Replikation ist Folgendes erforderlich:

- Der Outpost-CIDR-Zielbereich muss Ihrer Outpost-Quellsubnetztafel zugeordnet sein. Weitere Informationen finden Sie unter [Voraussetzungen für das Erstellen von Konfigurationsregeln](#).
- Für Quell- und Ziel-Buckets muss die S3-Versionsverwaltung aktiviert sein. Weitere Informationen über das Versioning finden Sie unter [Verwalten der S3-Versionierung für Ihren S3-on-Outposts-Bucket](#).
- Amazon S3 in Outposts muss über die Berechtigung verfügen, Objekte aus dem Quell-Bucket in Ihrem Namen in den Ziel-Bucket zu replizieren. Dies bedeutet, dass Sie eine Servicerolle zum Delegieren von GET- und PUT-Berechtigungen an S3 in Outposts erstellen müssen.
 1. Bevor Sie die Servicerolle erstellen, benötigen Sie die GET-Berechtigung für den Quell-Bucket und die PUT-Berechtigung für den Ziel-Bucket.
 2. Um die Servicerolle zum Delegieren von Berechtigungen an S3 in Outposts erstellen zu können, müssen Sie zunächst Berechtigungen konfigurieren, damit eine IAM-Entität (ein Benutzer oder eine Rolle) die Aktionen `iam:CreateRole` und `iam:PassRole` ausführen kann. Anschließend erlauben Sie der IAM-Entität, die Servicerolle zu erstellen. Damit S3 in Outposts die Servicerolle in Ihrem Namen annehmen kann und um GET- und PUT-Berechtigungen an S3 in Outposts zu delegieren, müssen Sie der Rolle die erforderlichen Vertrauens- und Berechtigungsrichtlinien zuordnen. Weitere Informationen zu diesen Berechtigungen für S3 in Outposts finden Sie unter [Erstellen einer IAM-Rolle](#). Weitere Informationen zum Erstellen einer Servicerolle finden Sie unter [Erstellen einer Servicerolle](#).

Was wird repliziert?

Standardmäßig repliziert S3 in Outposts Folgendes:

- Objekte, die nach dem Hinzufügen einer Replikations-Konfiguration erstellt wurden.

- Objektmetadaten von den Quellobjekten zu den Replikaten Informationen zum Replizieren von Metadaten aus den Replikaten zu den Quellobjekten finden Sie unter [Replikationsstatus, wenn die Synchronisierung von Amazon-S3-Replikatänderungen in Outposts aktiviert ist](#).
- Objekt-Markierungen, sofern vorhanden.

Auswirkungen von Löschvorgängen auf die Replikation

Wenn Sie ein Objekt aus dem Quell-Bucket löschen, werden standardmäßig die folgenden Aktionen ausgeführt:

- Wenn Sie eine DELETE-Anforderung ohne Angabe einer Objektversions-ID stellen, fügt S3 in Outposts eine Löschmarkierung hinzu. S3 in Outposts geht wie folgt mit der Löschmarkierung um:
 - S3 in Outposts repliziert die Löschmarkierung standardmäßig nicht.
 - Sie können jedoch die Löschmarkierungs-Replikation zu nicht Tag-basierten Regeln hinzufügen. Weitere Informationen zum Aktivieren der Löschmarkierungs-Replikation in Ihrer Replikations-Konfiguration finden Sie unter [Verwenden der S3-Konsole](#).
- Wenn Sie in einer DELETE-Anforderung eine zu löschende Objektversions-ID angeben, löscht S3 in Outposts diese Objektversion im Quell-Bucket dauerhaft. Die Löschung wird jedoch nicht in den Ziel-Buckets repliziert. Anders ausgedrückt: Dieselbe Objektversion wird aus den Ziel-Buckets nicht gelöscht. Dieses Verhalten schützt Daten vor böswilligen Löschungen.

Was wird nicht repliziert?

Standardmäßig repliziert S3 in Outposts Folgendes nicht:

- Objekte im Quell-Bucket, bei denen es sich um Replikate handelt, die von einer anderen Replikationsregel erstellt wurden. Zum Beispiel: Angenommen Sie konfigurieren eine Replikation, bei der Bucket A die Quelle und Bucket B das Ziel ist. Nehmen wir jetzt an, Sie fügen eine weitere Replikations-Konfiguration hinzu, bei der Bucket B die Quelle und Bucket C das Ziel ist. In diesem Fall werden Objekte in Bucket B, die Replikate von Objekten in Bucket A sind, nicht in Bucket C repliziert.
- Objekte im Quell-Bucket, die bereits auf ein anderes Ziel repliziert wurden. Wenn Sie beispielsweise den Ziel-Bucket in einer vorhandenen Replikations-Konfiguration ändern, repliziert S3 in Outposts diese Objekte nicht erneut.
- Objekte, die mit der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) erstellt wurden.

- Aktualisierungen von Unterressourcen auf Bucket-Ebene.

Wenn Sie beispielsweise die Lebenszyklus-Konfiguration ändern oder eine Benachrichtigungskonfiguration zu Ihrem Quell-Bucket hinzufügen, werden diese Änderungen nicht auf den Ziel-Bucket angewendet. Durch diese Funktion ist es möglich, für den Quell- und den Ziel-Bucket verschiedene Konfigurationen zu nutzen.

- Aktionen, die von der Lebenszyklus-Konfiguration durchgeführt werden.

Wenn Sie beispielsweise eine Lebenszykluskonfiguration nur auf Ihrem Quell-Bucket aktivieren und Ablaufaktionen konfigurieren, erstellt S3 in Outposts Löschmarkierungen für abgelaufene Objekte im Quell-Bucket, repliziert diese Markierungen jedoch nicht in den Ziel-Bucket. Wenn Sie dieselbe Lebenszyklus-Konfiguration sowohl auf den Quell- als auch auf den Ziel-Bucket anwenden möchten, aktivieren Sie für beide Buckets dieselbe Lebenszyklus-Konfiguration. Weitere Informationen zur Lebenszyklus-Konfiguration finden Sie unter [Verwaltung des Lebenszyklus von Objekten](#).

Was wird von S3 Replication in Outposts nicht unterstützt?

Die folgenden Funktionen von S3 Replication werden von S3 in Outposts derzeit nicht unterstützt:

- Begrenzung der S3-Replikationszeit (S3 RTC) S3 RTC wird nicht unterstützt, da der Objektdatenverkehr in S3 Replication in Outposts über Ihr On-Premises-Netzwerk (das lokale Gateway) übertragen wird. Weitere Informationen zu lokalen Gateways finden Sie unter [Arbeiten mit dem lokalen Gateway](#) im Benutzerhandbuch zu AWS Outposts.
- S3 Replication für Batchvorgänge.

Einrichten der Replikation

Note

Objekte, die bereits vor dem Einrichten einer Replikationsregel in Ihrem Bucket vorhanden waren, werden nicht automatisch repliziert. Anders ausgedrückt: Amazon S3 in Outposts repliziert Objekte nicht rückwirkend. Um Objekte zu replizieren, die vor der Konfiguration Ihrer Replikation erstellt wurden, können Sie diese unter Verwendung der API-Operation `CopyObject` in denselben Bucket kopieren. Nach dem Kopieren werden die Objekte als „neue“ Objekte im Bucket angezeigt und es gilt die Replikationskonfiguration für diese Objekte. Weitere Informationen zum Kopieren eines Objekts finden Sie unter [Kopieren eines](#)

[Objekte in einem Amazon S3 on Outposts-Bucket mit AWS SDK for Java](#) und [CopyObject](#) in Amazon Simple Storage Service – API-Referenz.

Um die S3 Replication in Outposts zu aktivieren, fügen Sie Ihrem Quell-Outposts-Bucket eine Replikationsregel hinzu. Die Replikationsregel weist S3 in Outposts an, Objekte wie angegeben zu replizieren. In der Replikationsregel müssen Sie Folgendes angeben:

- Den Zugriffspunkt des Quell-Outposts-Buckets – Den Amazon-Ressourcennamen (ARN) des Zugriffspunkts oder den Zugriffspunktalias des Buckets, von dem aus S3 in Outposts die Objekte replizieren soll. Weitere Informationen zur Verwendung von Zugriffspunktaliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-in-Outposts-Buckets](#).
- Die Objekte, die Sie replizieren möchten – Sie können alle Objekte im Quell-Outposts-Bucket replizieren oder nur eine Teilmenge davon. Teilmengen identifizieren Sie, indem Sie ein [Schlüsselnamenpräfix](#), mindestens ein Objekt-Tag oder beides in der Konfiguration angeben.

Wenn Sie beispielsweise eine Replikationsregel konfigurieren, um nur Objekte mit dem Schlüsselnamenpräfix `Tax/` zu replizieren, repliziert S3 in Outposts Objekte mit Schlüsseln wie `Tax/doc1` oder `Tax/doc2`. Es repliziert aber keine Objekte mit dem Schlüssel `Legal/doc3`. Wenn Sie sowohl ein Präfix als auch mindestens ein Tag angeben, repliziert S3 in Outposts nur Objekte, die dieses Schlüsselpräfix und diese Tags aufweisen.

- Den Ziel-Outposts-Bucket – Den ARN oder Zugriffspunktalias des Buckets, in den S3 in Outposts die Objekte replizieren soll.

Sie können die Replikationsregel über die REST-API, AWS-SDKs, die AWS Command Line Interface (AWS CLI) oder die Amazon-S3-Konsole konfigurieren.

S3 in Outposts stellt auch API-Vorgänge zur Unterstützung der Einrichtung von Replikationsregeln bereit. Weitere Informationen finden Sie in den folgenden Themen in der Amazon Simple Storage Service – API-Referenz.

- [PutBucketReplication](#)
- [GetBucketReplication](#)
- [DeleteBucketReplication](#)

Themen

- [Voraussetzungen für das Erstellen von Konfigurationsregeln](#)
- [Erstellen von Replikationsregeln in Outposts](#)

Voraussetzungen für das Erstellen von Konfigurationsregeln

Themen

- [Verbinden Ihrer Quell- und Ziel-Outpost-Subnetze](#)
- [Erstellen einer IAM-Rolle](#)

Verbinden Ihrer Quell- und Ziel-Outpost-Subnetze

Damit Ihr Replikationsdatenverkehr über Ihr lokales Gateway von Ihrem Quell-Outpost zu Ihrem Ziel-Outpost geleitet wird, müssen Sie eine neue Route hinzufügen, um das Netzwerk einzurichten. Sie müssen die Classless Inter-Domain Routing (CIDR)-Netzwerkbereiche Ihrer Zugriffspunkte miteinander verbinden. Für jedes Zugriffspunktpaar müssen Sie diese Verbindung nur einmal einrichten.

Einige Schritte zum Einrichten der Verbindung unterscheiden sich je nach Zugriffstyp Ihrer Outposts-Endpunkte, die Ihren Zugriffspunkten zugeordnet sind. Der Zugriffstyp für Endpunkte ist entweder Privat (direktes Virtual Private Cloud [VPC]-Routing für AWS Outposts) oder Kundeneigene IP-Adresse (ein kundeneigener IP-Adresspool [CoIP-Pool] in Ihrem On-Premises-Netzwerk).

Schritt 1: Ermitteln des CIDR-Bereichs Ihres Quell-Outposts-Endpunkts

So ermitteln Sie den CIDR-Bereich Ihres Quellendpunkts, der Ihrem Quellzugriffspunkt zugeordnet ist

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie in der Liste Outposts-Buckets den gewünschten Quell-Bucket für die Replikation aus.
4. Wählen Sie die Registerkarte Outposts-Zugriffspunkte und anschließend den Outposts-Zugriffspunkt für den Quell-Bucket für Ihre Replikationsregel aus.
5. Wählen Sie den Outposts-Endpunkt aus.
6. Kopieren Sie die Subnetz-ID, die in [Schritt 5](#) verwendet werden soll.
7. Die Methode, mit der Sie den CIDR-Bereich des Quell-Outposts-Endpunkts ermitteln, hängt vom Zugriffstyp Ihres Endpunkts ab.

Prüfen Sie im Abschnitt Outposts-Endpunkt – Übersicht den Zugriffstyp.

- Wenn der Zugriffstyp Privat lautet, kopieren Sie den Wert für Classless inter-domain routing (CIDR), der in [Schritt 6](#) verwendet werden soll.
- Wenn der Zugriffstyp Kundeneigene IP-Adresse lautet, gehen Sie wie folgt vor:
 1. Kopieren Sie den Wert für Kundeneigener IPv4-Pool, um ihn später als ID des Adresspools zu verwenden.
 2. Öffnen Sie die AWS Outposts-Konsole unter <https://console.aws.amazon.com/outposts/>.
 3. Wählen Sie im Navigationsbereich Lokale Gateway-Routing-Tabellen aus.
 4. Wählen Sie den Wert für Lokale Gateway-Routing-Tabellen-ID Ihres Quell-Outposts aus.
 5. Wählen Sie im Detailbereich die Registerkarte CoIP-Pools aus. Fügen Sie den Wert Ihrer CoIP-Pool-ID, den Sie zuvor kopiert haben, in das Suchfeld ein.
 6. Kopieren Sie für den übereinstimmenden CoIP-Pool den entsprechenden CIDRs-Wert Ihres Quell-Outposts-Endpunkts, um ihn in [Schritt 6](#) zu verwenden.

Schritt 2: Ermitteln der Subnetz-ID und des CIDR-Bereichs Ihres Ziel-Outposts-Endpunkts

Um die Subnetz-ID und den CIDR-Bereich Ihres Zielendpunkts zu ermitteln, der Ihrem Zielzugriffspunkt zugeordnet ist, führen Sie dieselben Unterschritte in [Schritt 1](#) aus und ändern Sie dabei Ihren Quell-Outposts-Endpunkt in Ihren Ziel-Outposts-Endpunkt. Kopieren Sie den Subnetz-ID-Wert Ihres Ziel-Outposts-Endpunkts, um ihn in [Schritt 6](#) zu verwenden. Kopieren Sie den CIDR-Wert Ihres Ziel-Outposts-Endpunkts, um ihn in [Schritt 5](#) zu verwenden.

Schritt 3: Ermitteln der lokalen Gateway-ID Ihres Quell-Outposts

So ermitteln Sie die lokale Gateway-ID Ihres Quell-Outposts

1. Öffnen Sie die AWS Outposts-Konsole unter <https://console.aws.amazon.com/outposts/>.
2. Wählen Sie im linken Navigationsbereich Lokale Gateways aus.
3. Suchen Sie auf der Seite Lokale Gateways nach der Outpost-ID Ihres Quell-Outposts, den Sie für die Replikation verwenden möchten.
4. Kopieren Sie den Wert der lokalen Gateway-ID Ihres Quell-Outposts, um ihn in [Schritt 5](#) zu verwenden.

Weitere Informationen zu lokalen Gateways finden Sie unter [Lokales Gateway](#) im AWS Outposts-Benutzerhandbuch.

Schritt 4: Ermitteln der lokalen Gateway-ID Ihres Ziel-Outposts

Um die lokale Gateway-ID Ihres Ziel-Outposts zu ermitteln, führen Sie dieselben Unterschritte in [Schritt 3](#) aus, wobei Sie allerdings nach der Outpost-ID für Ihren Ziel-Outpost suchen. Kopieren Sie den Wert der lokalen Gateway-ID Ihres Ziel-Outposts, um ihn in [Schritt 6](#) zu verwenden.

Schritt 5: Einrichten der Verbindung von Ihrem Quell-Outpost-Subnetz zu Ihrem Ziel-Outpost-Subnetz

So richten Sie eine Verbindung von Ihrem Quell-Outpost-Subnetz zu Ihrem Ziel-Outpost-Subnetz ein

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im linken Navigationsbereich die Option Subnets aus.
3. Geben Sie im Suchfeld die Subnetz-ID für Ihren Quell-Outposts-Endpunkt ein, den Sie in [Schritt 1](#) ermittelt haben. Wählen Sie das Subnetz mit der übereinstimmenden Subnetz-ID aus.
4. Wählen Sie für das übereinstimmende Subnetzelement den Wert für Routing-Tabelle dieses Subnetzes aus.
5. Wählen Sie auf der Seite mit ausgewählter Routing-Tabelle Aktionen und dann Routen bearbeiten aus.
6. Wählen Sie auf der Seite Routen bearbeiten die Option Route hinzufügen aus.
7. Geben Sie unter Ziel den CIDR-Bereich Ihres Ziel-Outposts-Endpunkts ein, den Sie in [Schritt 2](#) ermittelt haben.
8. Wählen Sie unter Ziel Outpost lokales Gateway aus und geben Sie die lokale Gateway-ID Ihres Quell-Outposts ein, die Sie in [Schritt 3](#) ermittelt haben.
9. Wählen Sie Änderungen speichern aus.
10. Vergewissern Sie sich, dass der Status für die Route Aktiv lautet.

Schritt 6: Einrichten der Verbindung von Ihrem Ziel-Outpost-Subnetz zu Ihrem Quell-Outpost-Subnetz

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im linken Navigationsbereich die Option Subnets aus.

3. Geben Sie im Suchfeld die Subnetz-ID für Ihren Ziel-Outposts-Endpunkt ein, den Sie in [Schritt 2](#) ermittelt haben. Wählen Sie das Subnetz mit der übereinstimmenden Subnetz-ID aus.
4. Wählen Sie für das übereinstimmende Subnetzelement den Wert für Routing-Tabelle dieses Subnetzes aus.
5. Wählen Sie auf der Seite mit ausgewählter Routing-Tabelle Aktionen und dann Routen bearbeiten aus.
6. Wählen Sie auf der Seite Routen bearbeiten die Option Route hinzufügen aus.
7. Geben Sie unter Ziel den CIDR-Bereich Ihres Ziel-Outposts-Endpunkts ein, den Sie in [Schritt 1](#) ermittelt haben.
8. Wählen Sie unter Ziel Outpost lokales Gateway aus und geben Sie die lokale Gateway-ID Ihres Ziel-Outposts ein, die Sie in [Schritt 4](#) ermittelt haben.
9. Wählen Sie Änderungen speichern aus.
10. Vergewissern Sie sich, dass der Status für die Route Aktiv lautet.

Nachdem Sie die CIDR-Netzwerkbereiche Ihrer Quell- und Zielzugriffspunkte verbunden haben, müssen Sie eine AWS Identity and Access Management (IAM)-Rolle erstellen.

Erstellen einer IAM-Rolle

Standardmäßig sind alle S3-in-Outputs-Ressourcen – Buckets, Objekte und zugehörige Unterressourcen – privat, sodass nur der Ressourcenbesitzer auf die Ressource zugreifen kann. S3 in Outputs benötigt Berechtigungen zum Lesen und Replizieren von Objekten aus dem Quell-Outposts-Bucket. Sie erteilen diese Berechtigungen, indem Sie eine IAM-Service-Rolle erstellen und die Rolle in der Replikationskonfiguration festlegen.

In diesem Abschnitt werden die Vertrauensrichtlinie und die mindestens erforderliche Berechtigungsrichtlinie erläutert. Diese Beispielanleitungen enthalten Schritt-für-Schritt-Anweisungen zum Erstellen einer IAM-Rolle. Weitere Informationen finden Sie unter [Erstellen von Replikationsregeln in Outposts](#). Weitere Informationen zu IAM-Rollen finden Sie unter [IAM-Rollen](#) im IAM-Benutzerhandbuch.

- Im folgenden Beispiel ist eine Vertrauensrichtlinie zu sehen, bei der Sie S3 in Outposts als Service-Prinzipal identifizieren, der die Rolle übernehmen kann.

```
{
  "Version": "2012-10-17",
  "Statement": [
```



```

    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3-outposts.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

- Im folgenden Beispiel wird eine Zugriffsrichtlinie gezeigt, bei der Sie der Rolle die Berechtigungen erteilen, Replikationsaufgaben in Ihrem Namen durchzuführen. Wenn S3 in Outposts die Rolle annimmt, verfügt es über die Berechtigungen, die Sie in dieser Richtlinie angeben. Wenn Sie diese Richtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Informationen. Stellen Sie sicher, dass Sie dabei die Outpost-IDs Ihrer Quell- und Ziel-Outposts sowie die Bucket-Namen und Zugriffspunktnamen Ihrer Quell- und Ziel-Outposts-Buckets verwenden.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3-outposts:GetObjectVersionForReplication",
        "s3-outposts:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3-outposts:ReplicateObject",
        "s3-outposts:ReplicateDelete"
      ],
      "Resource": [

```

```
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/  
bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",  
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/  
accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"  
    ]  
}  
]  
}
```

Die Zugriffsrichtlinie erteilt Berechtigungen für folgenden Aktionen:

- `s3-outposts:GetObjectVersionForReplication` – Die Berechtigung für diese Aktion wird für alle Objekte erteilt, damit S3 in Outposts eine bestimmte Objektversion abrufen kann, die jedem Objekt zugeordnet ist.
- `s3-outposts:GetObjectVersionTagging` – Die Berechtigung für diese Aktion für Objekte im `SOURCE-OUTPOSTS-BUCKET`-Bucket (Quell-Bucket) gestattet es S3 in Outposts, Objekt-Tags für die Replikation zu lesen. Weitere Informationen finden Sie unter [Hinzufügen von Tags für S3-on-Outposts-Buckets](#). Wenn S3 in Outposts nicht über diese Berechtigung verfügt, repliziert es die Objekte, nicht jedoch die Objekt-Tags.
- `s3-outposts:ReplicateObject` und `s3-outposts:ReplicateDelete` – Die Berechtigungen für diese Aktionen für alle Objekte im `DESTINATION-OUTPOSTS-BUCKET`-Bucket (Ziel-Bucket) erlauben es S3 in Outposts, Objekte oder Löschemarkierungen in den Ziel-Outposts-Bucket zu replizieren. Informationen zu Löschemarkierungen finden Sie unter [Auswirkungen von Löschemarkierungen auf die Replikation](#).

Note

- Die Berechtigung für die `s3-outposts:ReplicateObject`-Aktion im `DESTINATION-OUTPOSTS-BUCKET`-Bucket (Ziel-Bucket) erlaubt auch die Replikation von Objekt-Tags. Daher müssen Sie für die `s3-outposts:ReplicateTags`-Aktion keine explizite Berechtigung erteilen.
- Für die kontoübergreifende Replikation muss der Besitzer des Ziel-Outposts-Buckets seine Bucket-Richtlinie aktualisieren, um die Berechtigung für die `s3-outposts:ReplicateObject`-Aktion in dem `DESTINATION-OUTPOSTS-BUCKET` zu erteilen. Die `s3-outposts:ReplicateObject`-Aktion ermöglicht es S3 in Outposts, Objekte und Objekt-Tags in den Ziel-Outposts-Bucket zu replizieren.

Eine Liste der Aktionen von S3 in Outposts finden Sie unter [Aktionen, die von Amazon S3 in Outposts definiert werden](#).

⚠ Important

Das AWS-Konto, das die IAM-Rolle besitzt, muss über Berechtigungen für die Aktionen verfügen, die der IAM-Rolle erteilt werden.

Angenommen, der Quell-Outposts-Bucket enthält Objekte, die im Besitz eines anderen AWS-Kontos sind. Der Eigentümer der Objekte muss dem AWS-Konto, das die IAM-Rolle besitzt, die erforderlichen Berechtigungen explizit über die Bucket-Richtlinie und die Zugriffspunktrichtlinie erteilen. Andernfalls kann S3 in Outposts nicht auf die Objekte zugreifen und die Replikation der Objekte schlägt fehl.

Die hier beschriebenen Berechtigungen gehören zur Mindest-Replikationskonfiguration. Wenn Sie optionale Replikationskonfigurationen hinzufügen möchten, müssen Sie S3 in Outposts zusätzliche Berechtigungen erteilen.

Erteilen von Berechtigungen, wenn sich die Quell- und Ziel-Outposts-Buckets im Besitz verschiedener AWS-Konten befinden

Wenn sich die Quell- und Ziel-Outposts-Buckets nicht im Besitz desselben Kontos befinden, muss der Eigentümer des Ziel-Outposts-Buckets die Bucket- und Zugriffspunkt-Richtlinien für den Ziel-Bucket aktualisieren. Diese Richtlinien müssen dem Besitzer des Quell-Outposts-Buckets und der IAM-Servicerolle Berechtigungen zum Ausführen von Replikationsaktionen gewähren, wie in den folgenden Richtlinienbeispielen dargestellt. Andernfalls schlägt die Replikation fehl. In diesen Richtlinienbeispielen ist *DESTINATION-OUTPOSTS-BUCKET* der Ziel-Bucket. Wenn Sie diese Richtlinienbeispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

Wenn Sie die IAM-Servicerolle manuell erstellen, legen Sie den Rollenpfad als `role/service-role/` fest, wie in den folgenden Richtlinienbeispielen dargestellt. Weitere Informationen finden Sie unter [IAM ARNs](#) im IAM-Benutzerhandbuch.

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForDestinationBucket",
  "Statement": [
    {
```

```

        "Sid": "Permissions on objects",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
        },
        "Action": [
            "s3-outposts:ReplicateDelete",
            "s3-outposts:ReplicateObject"
        ],
        "Resource": [
            "arn:aws:s3-outposts:region:DestinationBucket-account-
ID:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*"
        ]
    }
]
}

```

```

{
  "Version": "2012-10-17",
  "Id": "PolicyForDestinationAccessPoint",
  "Statement": [
    {
      "Sid": "Permissions on objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
      },
      "Action": [
        "s3-outposts:ReplicateDelete",
        "s3-outposts:ReplicateObject"
      ],
      "Resource" : [
        "arn:aws:s3-outposts:region:DestinationBucket-account-
ID:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/
object/*"
      ]
    }
  ]
}

```


 Note

Wenn Objekte im Quell-Outposts-Bucket mit einem Tag versehen sind, beachten Sie Folgendes:

Wenn der Eigentümer des Quell-Outposts-Buckets S3 in Outposts die Berechtigung für die Aktionen `s3-outposts:GetObjectVersionTagging` und `s3-outposts:ReplicateTags` zum Replizieren von Objekt-Tags (über die IAM-Rolle) erteilt, repliziert Amazon S3 die Tags zusammen mit den Objekten. Weitere Information zur IAM-Rolle finden Sie unter [Erstellen einer IAM-Rolle](#).

Erstellen von Replikationsregeln in Outposts

Die S3-Replikation in Outposts ist eine automatische, asynchrone und Bucket-übergreifende Replikation von Objekten in demselben oder verschiedenen AWS Outposts. Bei der Replikation werden neu erstellte Objekte und Objektaktualisierungen aus einem Quell-Outposts-Bucket in einen oder mehrere Ziel-Outposts-Bucket(s) kopiert. Weitere Informationen finden Sie unter [Replikation von Objekten für S3 in Outposts](#).

 Note

Objekte, die bereits vor dem Einrichten von Replikationsregeln in Ihrem Quell-Outposts-Bucket vorhanden waren, werden nicht repliziert. Anders ausgedrückt: S3 in Outposts repliziert Objekte nicht rückwirkend. Um Objekte zu replizieren, die vor der Konfiguration Ihrer Replikation erstellt wurden, können Sie diese unter Verwendung der API-Operation `CopyObject` in denselben Bucket kopieren. Nach dem Kopieren werden die Objekte als „neue“ Objekte im Bucket angezeigt und es gilt die Replikationskonfiguration für diese Objekte. Weitere Informationen zum Kopieren eines Objekts finden Sie unter [Kopieren eines Objekts in einem Amazon S3 on Outposts-Bucket mit AWS SDK for Java](#) und [CopyObject in Amazon Simple Storage Service – API-Referenz](#).

Wenn Sie die Replikation konfigurieren, fügen Sie dem Quell-Outposts-Bucket Replikationsregeln hinzu. Replikationsregeln definieren, welche Quell-Outposts-Bucket-Objekte repliziert werden sollen und in welchem Ziel-Outposts-Bucket/welchen Ziel-Outposts-Buckets die replizierten Objekte gespeichert werden sollen. Sie können eine Regel erstellen, um alle Objekte in einem Bucket oder eine Untermenge von Objekten mit einem spezifischen Schlüsselnamenpräfixen, einem oder

mehreren Objekt-Markierungen oder beidem zu replizieren. Ein Ziel-Outposts-Bucket kann sich in demselben Outpost wie der Quell-Outposts-Bucket oder in einem anderen Outpost befinden.

Für die Replikationsregeln von S3 in Outposts müssen Sie sowohl den Amazon-Ressourcennamen (ARN) des Zugriffspunkts des Quell-Outposts-Buckets als auch den ARN des Zugriffspunkts des Ziel-Outposts-Buckets anstelle der Namen des Quell-Outposts-Buckets und des Ziel-Outposts-Buckets angeben.

Wenn Sie angeben, dass eine Objektversions-ID gelöscht werden soll, löscht S3 in Outposts diese Objektversion im Quell-Outposts-Bucket. Die Löschung wird jedoch nicht in den Ziel-Outposts-Bucket repliziert. Anders ausgedrückt: Dieselbe Objektversion wird nicht aus dem Ziel-Outposts-Bucket gelöscht. Dieses Verhalten schützt Daten vor böswilligen Löschungen.

Wenn Sie einem Outposts-Bucket eine Replikationsregel hinzufügen, ist diese standardmäßig aktiviert, sodass sie ausgeführt wird, sobald Sie sie speichern.

In diesem Beispiel richten Sie eine Replikation für Quell- und Ziel-Outposts-Buckets ein, die sich in unterschiedlichen Outposts befinden und demselben AWS-Konto gehören. Beispiele für die Verwendung der Amazon-S3-Konsole, der AWS Command Line Interface (AWS CLI) und der AWS SDK for Java und AWS SDK for .NET. Informationen zu den kontoübergreifenden Berechtigungen für die S3-Replikation in Outposts finden Sie unter [Erteilen von Berechtigungen, wenn sich die Quell- und Ziel-Outposts-Buckets im Besitz verschiedener AWS-Konten befinden](#).

Die Voraussetzungen für die Einrichtung der Replikationsregeln von S3 in Outposts finden Sie unter [Voraussetzungen für das Erstellen von Konfigurationsregeln](#).

Verwenden der S3-Konsole

Führen Sie die folgenden Schritte aus, um eine Replikationsregel zu konfigurieren, wenn sich der Amazon-S3-in-Outposts-Ziel-Bucket in einem anderen Outpost als der Quell-Outposts-Bucket befindet.

Wenn sich der Ziel-Outposts-Bucket in einem anderen Konto als der Quell-Outposts-Bucket befindet, müssen Sie dem Ziel-Outposts-Bucket eine Bucket-Richtlinie hinzufügen, um dem Eigentümer des Quell-Outposts-Bucket-Kontos die Berechtigung zum Replizieren von Objekten in den Ziel-Outposts-Bucket zu erteilen. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten](#).

So erstellen Sie eine Replikationsrolle

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie in der Liste Outposts-Buckets den Namen des Buckets aus, den Sie als Quell-Bucket verwenden möchten.
3. Wählen Sie die Registerkarte Verwaltung aus, scrollen Sie nach unten zum Abschnitt Replikationsregeln und wählen Sie dann Replikationsregel erstellen aus.
4. Geben Sie in Name der Replikationsregel einen Namen für Ihre Regel ein, um sie später besser identifizieren zu können. Der Name ist erforderlich und muss innerhalb des Buckets eindeutig sein.
5. In Status ist standardmäßig Aktiviert ausgewählt. Eine aktivierte Regel wird ausgeführt, sobald Sie speichern. Wenn Sie die Regel später aktivieren möchten, wählen Sie Deaktiviert aus.
6. Unter Priorität bestimmt der Prioritätswert der Regel, welche Regel im Falle einer Überschneidung von Regeln angewendet wird. Wenn Objekte in den Geltungsbereich von mehr als einer Replikationsregel fallen, verwendet S3 in Outposts diese Prioritätswerte, um Konflikte zu vermeiden. Standardmäßig werden neue Regeln mit der höchsten Priorität zur Replikationskonfiguration hinzugefügt. Je höher die Zahl, desto höher die Priorität.

Um die Priorität für die Regel zu ändern, wählen Sie nach dem Speichern der Regel zunächst den Namen der Regel aus der Liste der Replikationsregeln, dann Aktionen und schließlich Priorität bearbeiten aus.

7. Unter Quell-Bucket stehen Ihnen folgende Optionen zum Festlegen der Replikationsquelle zur Verfügung:
 - Um den gesamten Bucket zu replizieren, wählen Sie Auf alle Objekte im Bucket anwenden aus.
 - Um die Präfix- oder Tag-Filterung auf die Replikationsquelle anzuwenden, wählen Sie Geltungsbereich dieser Regel durch Verwendung von einem oder mehreren Filtern beschränken aus. Sie können ein Präfix und Markierungen kombinieren.
 - Um alle Objekte mit demselben Präfix zu replizieren, geben Sie unter Präfix ein Präfix in das Feld ein. Bei Verwendung des Filters Präfix ist die Replikation auf alle Objekte beschränkt, deren Namen mit derselben Zeichenfolge beginnen (z. B. `pictures`).

Wenn Sie ein Präfix eingeben, bei dem es sich um den Namen eines Ordners handelt, müssen Sie einen / (Schrägstrich) als letztes Zeichen eingeben (z. B. `pictures/`).

- Um alle Objekte mit einem oder mehreren gleichen Objekt-Tags zu replizieren, wählen Sie Tag hinzufügen aus und geben Sie das Schlüssel-Wert-Paar in die Felder ein. Wiederholen Sie den Vorgang, um ein weiteres Tag hinzuzufügen. Weitere Informationen über Objekt-Markierungen finden Sie unter [Hinzufügen von Tags für S3-on-Outposts-Buckets](#).
8. Um für die Replikation auf Ihren S3-in-Outposts-Quell-Bucket zuzugreifen, wählen Sie unter Quellzugriffspunktname einen Zugriffspunkt aus, der an den Quell-Bucket angehängt ist.
 9. Wählen Sie unter Ziel den Zugriffspunkt-ARN des Ziel-Outposts-Buckets aus, in den S3 in Outposts Objekte replizieren soll. Der Ziel-Outposts-Bucket kann sich in demselben AWS-Konto wie der Quell-Outposts-Bucket oder in einem anderen befinden.

Wenn sich der Ziel-Bucket in einem anderen Konto als der Quell-Outposts-Bucket befindet, müssen Sie dem Ziel-Outposts-Bucket eine Bucket-Richtlinie hinzufügen, um dem Eigentümer des Quell-Outposts-Bucket-Kontos die Berechtigung zum Replizieren von Objekten in den Ziel-Outposts-Bucket zu erteilen. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen, wenn sich die Quell- und Ziel-Outposts-Buckets im Besitz verschiedener AWS-Konten befinden](#).

Note

Wenn die Versionsverwaltung für den Ziel-Outposts-Bucket nicht aktiviert ist, erhalten Sie eine Warnmeldung, die die Schaltfläche Versionierung aktivieren enthält. Wählen Sie diese Schaltfläche, um das Versioning für den Bucket zu aktivieren.

10. Richten Sie eine AWS Identity and Access Management (IAM)-Servicerolle ein, die S3 in Outposts annehmen kann, um Objekte in Ihrem Namen zu replizieren.

Führen Sie zum Einrichten einer IAM-Rolle unter IAM-Rolle einen der folgenden Schritte aus:

- Damit S3 in Outposts eine neue IAM-Rolle für Ihre Replikationskonfiguration erstellt, wählen Sie Aus vorhandenen IAM-Rollen auswählen und dann Neue Rolle erstellen aus. Wenn Sie die Regel speichern, wird eine neue Richtlinie für die IAM-Rolle erstellt, die mit den von Ihnen ausgewählten Quell- und Ziel-Outposts-Buckets übereinstimmt. Wir empfehlen Ihnen, die Option Neue Rolle erstellen auszuwählen.
- Sie haben auch die Möglichkeit, eine vorhandene IAM-Rolle zu verwenden. In diesem Fall müssen Sie eine Rolle auswählen, die S3 in Outposts die erforderlichen Berechtigungen für die Replikation gewährt. Wenn diese Rolle S3 in Outposts keine ausreichenden Berechtigungen gewährt, um Ihre Replikationsregel zu befolgen, schlägt die Replikation fehl.

Um eine vorhandene Rolle auszuwählen, wählen Sie Aus vorhandenen IAM-Rollen auswählen und dann im Dropdown-Menü die Rolle aus. Sie können auch die Option IAM-Rollen-ARN eingeben auswählen und dann den Amazon-Ressourcennamen (ARN) der IAM-Rolle eingeben.

⚠ Important

Wenn Sie eine Replikationsregel zu einem S3-in-Outposts-Bucket hinzufügen, benötigen Sie die Berechtigungen `iam:CreateRole` und `iam:PassRole`, um die IAM-Rolle erstellen und übergeben zu können, die S3 in Outposts Replikationsberechtigungen gewährt. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Service übergeben kann](#) im IAM-Benutzerhandbuch.

11. Alle Objekte in Outposts-Buckets sind standardmäßig verschlüsselt. Weitere Informationen zur Verschlüsselung in S3 in Outposts finden Sie unter [Datenverschlüsselung in S3 on Outposts](#). Nur Objekte, die durch die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) verschlüsselt wurden, können repliziert werden. Die Replikation von Objekten, die durch serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS)-Schlüsseln (SSE-KMS) oder durch serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, wird nicht unterstützt.
12. Aktivieren Sie beim Festlegen der Konfiguration der Replikationsregeln nach Bedarf die folgenden zusätzlichen Optionen:
 - Wenn Sie S3-in-Outposts-Replikationsmetriken in Ihrer Replikationskonfiguration aktivieren möchten, wählen Sie Replikationsmetriken aus. Weitere Informationen finden Sie unter [Überwachen des Fortschritts mit Replikationsmetriken](#).
 - Wenn Sie die Löschmarkierungs-Replikation in Ihrer Replikations-Konfiguration aktivieren möchten, wählen Sie Markierungsreplikation löschen aus. Weitere Informationen finden Sie unter [Auswirkungen von Löschvorgängen auf die Replikation](#).
 - Wenn Sie an den Replikaten vorgenommene Metadatenänderungen zurück in die Quellobjekte replizieren möchten, wählen Sie Synchronisierung von Replikatänderungen aus. Weitere Informationen finden Sie unter [Replikationsstatus, wenn die Synchronisierung von Amazon-S3-Replikatänderungen in Outposts aktiviert ist](#).
13. Wählen Sie Regel erstellen aus, um den Vorgang abzuschließen.

Nach dem Speichern der Regel können Sie diese bearbeiten, aktivieren, deaktivieren oder löschen. Wechseln Sie hierfür auf die Registerkarte Verwaltung für den Quell-Outposts-Bucket, scrollen Sie nach unten zum Abschnitt Replikationsregeln, wählen Sie Ihre Regel aus und wählen Sie dann Regel bearbeiten aus.

Verwendung von AWS CLI

Um die AWS CLI zum Einrichten der Replikation zu verwenden, wenn sich die Quell- und Ziel-Outposts-Buckets im Besitz desselben AWS-Kontos befinden, gehen Sie wie folgt vor:

- Erstellen Sie Quell- und Ziel-Outposts-Buckets.
- Aktivieren Sie die Versionsverwaltung für beide Buckets.
- Erstellen Sie eine IAM-Rolle, die S3 in Outposts die Berechtigung zur Replikation von Objekten erteilt.
- Fügen Sie die Replikationskonfiguration zum Quell-Outposts-Bucket hinzu.

Um die Einrichtung zu prüfen, testen Sie sie.

So richten Sie die Replikation ein, wenn sich die Quell- und Ziel-Outposts-Buckets im Besitz desselben AWS-Kontos befinden

1. Richten Sie das Anmeldeinformationsprofil für die AWS CLI ein. In diesem Beispiel verwenden wir den Profilnamen `acctA`. Informationen zum Einrichten der Anmeldeinformations-Profile finden Sie unter [Named Profiles](#) (Benannte Profile) im AWS Command Line Interface-Benutzerhandbuch.

Important

Das Profil, das Sie für diese Übung verwenden, muss über die nötigen Berechtigungen verfügen. Beispielsweise legen Sie in der Replikationskonfiguration die IAM-Servicerolle fest, die S3 in Outposts annehmen kann. Dies können Sie nur tun, wenn das verwendete Profil über die Berechtigungen `iam:CreateRole` und `iam:PassRole` verfügt. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Service übergeben kann](#) im IAM-Benutzerhandbuch. Wenn Sie zur Erstellung eines benannten Profils die Anmeldeinformationen eines Administrators verwenden, verfügt das benannte Profil über die erforderliche Berechtigung, um alle Aufgaben durchzuführen.

- Erstellen Sie einen Quell-Bucket und aktivieren Sie das Versioning für ihn. Mit dem folgenden Befehl `create-bucket` wird ein ***SOURCE-OUTPOSTS-BUCKET***-Bucket in der Region USA Ost (Nord-Virginia) (`us-east-1`) erstellt. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws s3control create-bucket --bucket SOURCE-OUTPOSTS-BUCKET --outpost-id SOURCE-OUTPOST-ID --profile acctA --region us-east-1
```

Mit dem folgenden Befehl `put-bucket-versioning` wird die Versionsverwaltung auf dem ***SOURCE-OUTPOSTS-BUCKET***-Bucket aktiviert. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

- Erstellen Sie einen Ziel-Bucket und aktivieren Sie das Versioning für ihn. Mit dem folgenden Befehl `create-bucket` wird ein ***DESTINATION-OUTPOSTS-BUCKET***-Bucket in der Region USA West (Oregon) (`us-west-2`) erstellt. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

Note

Um die Replikationskonfiguration einzurichten, wenn sich die Quell- und Ziel-Outposts-Bucket in demselben AWS-Konto befinden, verwenden Sie dasselbe benannte Profil. Dieses Beispiel verwendet `acctA`. Zum Testen der Replikationskonfiguration, wenn sich die Buckets im Besitz unterschiedlicher AWS-Konten befinden, legen Sie verschiedene Profile für jeden Bucket fest.

```
aws s3control create-bucket --bucket DESTINATION-OUTPOSTS-BUCKET --create-bucket-configuration LocationConstraint=us-west-2 --outpost-id DESTINATION-OUTPOST-ID --profile acctA --region us-west-2
```

Mit dem folgenden Befehl `put-bucket-versioning` wird die Versionsverwaltung auf dem ***DESTINATION-OUTPOSTS-BUCKET***-Bucket aktiviert. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

4. Erstellen Sie eine IAM-Servicerolle. Zu einem späteren Zeitpunkt der Replikationskonfiguration fügen Sie diese Servicerolle dem *SOURCE-OUTPOSTS-BUCKET*-Bucket hinzu. S3 in Outposts übernimmt diese Rolle, um Objekte in Ihrem Namen zu replizieren. Sie erstellen eine IAM-Rolle in zwei Schritten:

- a. Erstellen Sie eine IAM-Rolle.

- i. Kopieren Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-on-outposts-role-trust-policy.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie gewährt S3 in Outposts Service-Prinzipal-Berechtigungen, um die Servicerolle anzunehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3-outposts.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- ii. Führen Sie den folgenden -Befehl aus, um die Rolle zu erstellen. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws iam create-role --role-name replicationRole --assume-role-policy-document file://s3-on-outposts-role-trust-policy.json --profile acctA
```

- b. Fügen Sie eine Berechtigungsrichtlinie zur Servicerolle hinzu.

- i. Kopieren Sie die folgende Berechtigungsrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-on-outposts-role-permissions-policy.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie erteilt Berechtigungen für

verschiedene S3-in-Outposts-Bucket- und -Objektaktionen. Wenn Sie diese Richtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Informationen.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3-outposts:GetObjectVersionForReplication",
        "s3-outposts:GetObjectVersionTagging"
      ],
      "Resource":[
        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3-outposts:ReplicateObject",
        "s3-outposts:ReplicateDelete"
      ],
      "Resource":[
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
      ]
    }
  ]
}
```

- ii. Führen Sie den folgenden Befehl aus, um eine Richtlinie zu erstellen und sie der Rolle anzufügen. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws iam put-role-policy --role-name replicationRole --policy-document file://s3-on-outposts-role-permissions-policy.json --policy-name replicationRolePolicy --profile acctA
```

5. Fügen Sie eine Replikationskonfiguration zum *SOURCE-OUTPOSTS-BUCKET*-Bucket hinzu.

- a. Zwar erfordert die S3-in-Outposts-API eine Replikationskonfiguration im XML-Format, die AWS CLI verlangt jedoch die Angabe der Replikationskonfiguration im JSON-Format. Speichern Sie den folgenden JSON-Code in einer Datei mit dem Namen `replication.json` im lokalen Verzeichnis auf Ihrem Computer. Wenn Sie diese Konfiguration verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket":
          "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT"
      }
    }
  ]
}
```


- b. Führen Sie den folgenden Befehl `put-bucket-replication` aus, um die Replikationskonfiguration zu Ihrem Quell-Outposts-Bucket hinzuzufügen. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws s3control put-bucket-replication --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET --replication-configuration file://replication.json --profile acctA
```

- c. Um die Replikationskonfiguration abzurufen, verwenden Sie den Befehl `get-bucket-replication`. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws s3control get-bucket-replication --account-id 123456789012 --bucket
arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
bucket/SOURCE-OUTPOSTS-BUCKET --profile acctA
```

6. Testen Sie das Setup in der Amazon-S3-Konsole:
 - a. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
 - b. Erstellen Sie im *SOURCE-OUTPOSTS-BUCKET*-Bucket einen Ordner mit dem Namen Tax.
 - c. Fügen Sie Beispielobjekte zum Tax-Ordner im *SOURCE-OUTPOSTS-BUCKET*-Bucket hinzu.
 - d. Überprüfen Sie im *DESTINATION-OUTPOSTS-BUCKET*-Bucket Folgendes:
 - S3 in Outposts hat die Objekte repliziert.

 Note

Die von S3 in Outposts für die Replikation eines Objekts benötigte Zeit hängt von der Größe des Objekts ab. Weitere Informationen zum Anzeigen des Replikationsstatus finden Sie unter [Abrufen von Replikationsstatusinformationen](#).

- Auf der Registerkarte Eigenschaften des Objekts ist Replikationsstatus auf Replikat gesetzt (sodass dies als Replikatobjekt identifiziert wird).

Verwalten Ihrer Replikation

In diesem Abschnitt werden zusätzliche Optionen für die Replikationskonfiguration beschrieben, die in S3 in Outposts verfügbar sind, und es wird erörtert, wie Sie den Replikationsstatus ermitteln und Replikationsprobleme beheben können. Weitere Informationen zum Erstellen einer grundlegenden Replikationskonfiguration finden Sie unter [Einrichten der Replikation](#).

Themen

- [Überwachen des Fortschritts mit Replikationsmetriken](#)
- [Abrufen von Replikationsstatusinformationen](#)
- [Fehlerbehebung bei einer Replikation](#)
- [Verwenden von EventBridge für S3 Replication in Outposts](#)

Überwachen des Fortschritts mit Replikationsmetriken

S3 Replication in Outposts bietet detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Mithilfe der Replikationsmetriken können Sie den Fortschritt der Replikation in 5-Minuten-Intervallen überwachen. Verfolgen Sie dazu die Bytes der ausstehenden Replikation, die Replikationslatenz und die Operationen mit ausstehender Replikation. Zur Unterstützung bei der Behebung von Konfigurationsproblemen können Sie Amazon EventBridge auch so einrichten, dass Nachrichten über Replikationsfehler erhalten werden.

Wenn Replikationsmetriken aktiviert sind, veröffentlicht S3 Replication in Outposts die folgenden Metriken in Amazon CloudWatch:

- Bytes der ausstehenden Replikation – Die Gesamtzahl der Bytes von Objekten, deren Replikation für eine bestimmte Replikationsregel aussteht.
- Replikationslatenz – Die maximale Anzahl von Sekunden, um die der Replikations-Ziel-Bucket für eine bestimmte Replikationsregel hinter dem Quell-Bucket zurückliegt.
- Operationen mit ausstehender Replikation – Die Anzahl der Operationen, deren Replikation für eine bestimmte Replikationsregel aussteht. Zu den Operationen gehören Objekte, Löschmarkierungen und Tags.

Note

Die Metriken von S3 Replication in Outposts werden zu demselben Preis abgerechnet wie benutzerdefinierte CloudWatch-Metriken. Weitere Informationen hierzu finden Sie unter [Amazon CloudWatch – Preise](#).

Abrufen von Replikationsstatusinformationen

Der Replikationsstatus hilft Ihnen, den aktuellen Status eines Objekts zu bestimmen, das gerade von Amazon S3 in Outposts repliziert wird. Der Replikationsstatus eines Quellobjekts gibt entweder PENDING, COMPLETED oder FAILED zurück. Der Replikationsstatus eines Replikats gibt REPLICA zurück.

Übersicht über den Replikationsstatus

In einem Replikationsszenario haben Sie einen Quell-Bucket, auf dem Sie die Replikation konfigurieren, und einen Ziel-Bucket, in den S3 in Outposts Objekte repliziert. Wenn Sie ein Objekt

(unter Verwendung von `GetObject`) oder Objektmetadaten (unter Verwendung von `HeadObject`) von diesen Buckets anfordern, gibt S3 in Outposts den Header `x-amz-replication-status` wie folgt in der Antwort zurück:

- Wenn Sie ein Objekt aus dem Quell-Bucket anfordern, gibt S3 in Outposts den Header `x-amz-replication-status` zurück, wenn das Objekt in der Anforderung für die Replikation geeignet ist.

Nehmen wir beispielsweise an, dass Sie in Ihrer Replikationskonfiguration das Objektpräfix `TaxDocs` angeben, um S3 in Outposts anzuweisen, nur Objekte mit dem Schlüsselnamenpräfix `TaxDocs` zu replizieren. Alle Objekte mit diesem Schlüsselnamenpräfix, die Sie hochladen, z. B. `TaxDocs/document1.pdf`, werden repliziert. Für Objektanforderungen mit diesem Schlüsselnamenpräfix gibt S3 in Outposts den Header `x-amz-replication-status` mit einem der folgenden Werte für den Replikationsstatus des Objekts zurück: `PENDING`, `COMPLETED` oder `FAILED`.

Note

Wenn nach dem Hochladen eines Objekts die Objektreplikation fehlschlägt, können Sie die fehlgeschlagene Replikation nicht erneut durchzuführen versuchen. Sie müssen das Objekt erneut hochladen. Bei Problemen wie fehlenden Replikationsrollen-Berechtigungen oder fehlenden Bucket-Berechtigungen gehen Objekte in den Status `FAILED` über. Bei temporären Fehlern, z. B. wenn ein Bucket oder Ihr Outpost nicht verfügbar ist, geht der Replikationsstatus nicht in `FAILED` über, sondern verbleibt bei `PENDING`. Wenn die Ressource wieder online ist, setzt S3 in Outposts die Replikation dieser Objekte fort.

- Wenn Sie ein Objekt aus einem Ziel-Bucket anfordern und es sich bei dem Objekt in Ihrer Anforderung um ein Replikat handelt, das von S3 in Outposts erstellt wurde, gibt S3 in Outposts den Header `x-amz-replication-status` mit dem Wert `REPLICA` zurück.

Note

Bevor Sie ein Objekt aus einem Quell-Bucket löschen, bei dem die Replikation aktiviert ist, sollten Sie den Replikationsstatus des Objekts überprüfen, um sicherzustellen, dass das Objekt repliziert wurde.

Replikationsstatus, wenn die Synchronisierung von Amazon-S3-Replikatänderungen in Outposts aktiviert ist

Wenn in Ihren Replikationsregeln die Synchronisierung von S3-in-Outposts-Replikatänderungen aktiviert ist, können Replikate einen anderen Status als REPLICIA melden. Wenn Änderungen an Metadaten gerade repliziert werden, gibt der Header `x-amz-replication-status` für das Replikat PENDING zurück. Wenn bei der Synchronisierung der Replikatänderungen die Replikation von Metadaten fehlschlägt, gibt der Header für das Replikat FAILED zurück. Wenn Metadaten korrekt repliziert werden, gibt der Header für das Replikat den Wert REPLICIA zurück.

Fehlerbehebung bei einer Replikation

Wenn Objektreplikate nicht im Amazon-S3-in-Outposts-Ziel-Bucket angezeigt werden, nachdem Sie die Replikation konfiguriert haben, können Sie mit diesen Tipps zur Fehlerbehebung Probleme identifizieren und beheben.

- Wie lange Amazon S3 in Outposts für die Replikation eines Objekts benötigt, hängt von verschiedenen Faktoren ab, unter anderem von der Distanz zwischen den Quell- und Ziel-Outposts und der Größe des Objekts.

Sie können den Replikationsstatus des Quellobjekts überprüfen. Wenn der Replikationsstatus des Objekts PENDING lautet, hat S3 in Outposts die Replikation noch nicht abgeschlossen. Wenn der Replikationsstatus des Objekts FAILED lautet, überprüfen Sie die Replikationskonfiguration des Quell-Buckets.

- Überprüfen Sie in der Replikations-Konfiguration des Quell-Buckets Folgendes:
 - ob der Amazon-Ressourcenname (ARN) des Zugriffspunkts des Ziel-Buckets korrekt ist.
 - ob das Schlüsselnamenpräfix korrekt ist. Wenn Sie beispielsweise die Konfiguration so einrichten, dass nur Objekte mit dem Präfix `Tax` repliziert werden, werden nur Objekte mit Schlüsselnamen wie beispielsweise `Tax/document1` oder `Tax/document2` repliziert. Ein Objekt mit dem Schlüsselnamen `document3` wird nicht repliziert.
 - Der Status lautet `Enabled`.
- Stellen Sie sicher, dass die Versionsverwaltung bei keinem der beiden Buckets ausgesetzt wurde. Sowohl für die Quell- als auch für die Ziel-Buckets muss die Versionsverwaltung aktiviert sein.
- Wenn sich der Ziel-Bucket im Besitz eines anderen AWS-Konto befindet, stellen Sie sicher, dass der Bucket-Eigentümer eine Bucket-Richtlinie für den Ziel-Bucket eingerichtet hat, die dem Eigentümer des Quell-Buckets die Replikation von Objekten gestattet. Ein Beispiel finden Sie

unter [Erteilen von Berechtigungen, wenn sich die Quell- und Ziel-Outposts-Buckets im Besitz verschiedener AWS-Konten befinden](#).

- Wenn ein Objektreplikat nicht im Ziel-Bucket angezeigt wird, könnte Folgendes die Replikation verhindern:
 - S3 in Outposts repliziert keine Objekte in einem Quell-Bucket, bei dem es sich um ein Replikat handelt, das mit einer anderen Replikationskonfiguration erstellt wurde. Wenn Sie beispielsweise ein Replikationskonfiguration von Bucket A zu Bucket B zu Bucket C festlegen, repliziert S3 in Outposts keine Objektreplicate in Bucket B zu Bucket C.

Wenn Sie Objekte in Bucket A zu Bucket B und Bucket C replizieren möchten, legen Sie mehrere Bucket-Ziele in unterschiedlichen Replikationsregeln für Ihre Quell-Bucket-Replikationskonfiguration fest. Erstellen Sie beispielsweise zwei Replikationsregeln für Quell-Bucket A, wobei eine Regel für die Replikation in Ziel-Bucket B und die andere Regel für die Replikation in Ziel-Bucket C gilt.

- Ein Quell-Bucket-Eigentümer kann anderen AWS-Konten Berechtigungen für das Hochladen von Objekten erteilen. Standardmäßig besitzt der Quell-Bucket-Eigentümer keine Berechtigungen für die Objekte, die von anderen Konten erstellt wurden. Die Replikations-Konfiguration repliziert nur die Objekte, für die der Quell-Bucket-Eigentümer über Zugriffsberechtigungen verfügt. Um Replikationsprobleme zu vermeiden, kann der Quell-Bucket-Eigentümer anderen AWS-Konten Berechtigungen zum bedingten Erstellen von Objekten erteilen. Dabei sind explizite Zugriffsberechtigungen für diese Objekte erforderlich. Eine Beispielrichtlinie finden Sie unter [Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt](#).
- Angenommen, Sie fügen einer Replikationskonfiguration eine Regel hinzu, um eine Teilmenge von Objekten mit einem spezifischen Tag zu replizieren. In diesem Fall müssen Sie den spezifischen Tag-Schlüssel und -Wert zum Zeitpunkt der Objekterstellung zuweisen, damit S3 in Outposts das Objekt replizieren kann. Wenn Sie zuerst ein Objekt erstellen und dann dem vorhandenen Objekt das Tag hinzufügen, repliziert S3 in Outposts das Objekt nicht.
- Die Replikation schlägt fehl, wenn die Bucket-Richtlinie den Zugriff auf die Replikationsrolle für eine der folgenden Aktionen verweigert:

Quell-Bucket

```
"s3-outposts:GetObjectVersionForReplication",  
"s3-outposts:GetObjectVersionTagging"
```

Ziel-Buckets:

```
"s3-outposts:ReplicateObject",  
"s3-outposts:ReplicateDelete",  
"s3-outposts:ReplicateTags"
```

- Amazon EventBridge kann Sie darüber informieren, wenn Objekte nicht in ihre Ziel-Outposts repliziert werden. Weitere Informationen finden Sie unter [Verwenden von EventBridge für S3 Replication in Outposts](#).

Verwenden von EventBridge für S3 Replication in Outposts

Amazon S3 in Outposts ist in Amazon EventBridge integriert und verwendet den Namespace `s3-outposts`. EventBridge ist ein Serverless-Event-Bus-Service, mit dem Sie Ihre Anwendungen mit Daten aus verschiedenen Quellen verbinden können. Weitere Informationen finden Sie unter [Was ist Amazon EventBridge?](#) im Amazon-EventBridge-Benutzerhandbuch.

Zur Unterstützung bei der Behebung von Problemen mit der Replikationskonfiguration können Sie Amazon EventBridge so einrichten, dass Nachrichten über Replikationsfehlerereignisse erhalten werden. EventBridge kann Sie darüber informieren, wenn Objekte nicht in ihre Ziel-Outposts repliziert werden. Weitere Informationen zum aktuellen Status eines zu replizierenden Objekts finden Sie unter [Übersicht über den Replikationsstatus](#).

S3 in Outposts kann Ereignisse an EventBridge senden, wenn bestimmte Ereignisse in Ihrem Outposts-Bucket stattfinden. Anders als bei anderen Zielen müssen Sie nicht auswählen, welche Ereignistypen Sie liefern möchten. Sie können EventBridge-Regeln auch verwenden, um Ereignisse an zusätzliche Ziele weiterzuleiten. Nachdem EventBridge aktiviert wurde, sendet S3 in Outposts alle folgenden Ereignisse an EventBridge.

Ereignistyp	Beschreibung	Namespace
OperationFailedReplication	Die Replikation eines Objekts innerhalb einer Replikationsregel ist fehlgeschlagen. Weitere Informationen darüber, warum S3 Replication in Outposts fehlgeschlagen ist, finden Sie unter Verwenden von EventBrid	s3-outposts

Ereignistyp	Beschreibung	Namespace
	ge zur Anzeige der Ursachen von Fehlern bei S3 Replication in Outposts.	

Verwenden von EventBridge zur Anzeige der Ursachen von Fehlern bei S3 Replication in Outposts

In der folgenden Tabelle sind Gründe für das Fehlschlagen von S3 Replication in Outposts aufgeführt. Sie können eine EventBridge-Regel so konfigurieren, dass die Fehlerursache über Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), AWS Lambda oder Amazon CloudWatch Logs veröffentlicht und angezeigt wird. Weitere Informationen zu den erforderlichen Berechtigungen zur Verwendung dieser Ressourcen für EventBridge finden Sie unter [Verwenden von ressourcenbasierten Richtlinien für EventBridge](#).

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>AssumeRoleNotPermitted</code>	S3 in Outposts kann die in der Replikationskonfiguration angegebene AWS Identity and Access Management (IAM)-Rolle nicht übernehmen.
<code>DstBucketNotFound</code>	S3 in Outposts kann den in der Replikationskonfiguration angegebenen Ziel-Bucket nicht finden.
<code>DstBucketUnversioned</code>	Die Versionsverwaltung ist im Outposts-Ziel-Bucket nicht aktiviert. Um Objekte mit S3 Replication in Outposts replizieren zu können, müssen Sie die Versionsverwaltung im Ziel-Bucket aktivieren.
<code>DstDelObjNotPermitted</code>	S3 in Outposts kann Löschvorgänge nicht in den Ziel-Bucket replizieren. Möglicherweise fehlt die <code>s3-outposts:ReplicateDelete</code> -Berechtigung für den Ziel-Bucket.

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>DstMultipartCompleteNotPermitted</code>	S3 in Outposts kann einen mehrteiligen Upload von Objekten in den Ziel-Bucket nicht abschließen. Möglicherweise fehlt die <code>s3-outposts:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstMultipartInitNotPermitted</code>	S3 in Outposts kann einen mehrteiligen Upload von Objekten in den Ziel-Bucket nicht initiieren. Möglicherweise fehlt die <code>s3-outposts:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstMultipartPartUploadNotPermitted</code>	S3 in Outposts kann keine mehrteiligen Objekte in den Ziel-Bucket hochladen. Möglicherweise fehlt die <code>s3-outposts:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstOutOfCapacity</code>	S3 in Outposts kann nicht in den Ziel-Outpost replizieren, da die S3-Speicherkapazität des Outposts aufgebraucht ist.
<code>DstPutObjNotPermitted</code>	S3 in Outposts kann keine Objekte in den Ziel-Bucket replizieren. Möglicherweise fehlt die <code>s3-outposts:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstPutTaggingNotPermitted</code>	S3 in Outposts kann keine Objekt-Tags in den Ziel-Bucket replizieren. Möglicherweise fehlt die <code>s3-outposts:ReplicateObject</code> -Berechtigung für den Ziel-Bucket.
<code>DstVersionNotFound</code>	S3 in Outposts kann die Objektversion, die benötigt wird, um die Metadaten dieser Objektversion zu replizieren, nicht im Ziel-Bucket finden.

Gründe für das Fehlschlagen der Replikation	Beschreibung
<code>SrcBucketReplicationConfigMissing</code>	S3 in Outposts kann keine Replikationskonfiguration für den Zugriffspunkt finden, der dem Quell-Outposts-Bucket zugeordnet ist.
<code>SrcGetObjectNotPermitted</code>	S3 in Outposts kann nicht auf das Objekt im Quell-Bucket für die Replikation zugreifen . Möglicherweise fehlt die <code>s3-outposts:GetObjectVersionForReplication</code> -Berechtigung für den Quell-Bucket.
<code>SrcGetTaggingNotPermitted</code>	S3 in Outposts kann nicht auf Objekt-Tag-Informationen vom Quell-Bucket zugreifen . Möglicherweise fehlt die <code>s3-outposts:GetObjectVersionTagging</code> -Berechtigung für den Quell-Bucket.
<code>SrcHeadObjectNotPermitted</code>	S3 in Outposts kann keine Objektmetadaten aus dem Quell-Bucket abrufen. Möglicherweise fehlt die <code>s3-outposts:GetObjectVersionForReplication</code> -Berechtigung für den Quell-Bucket.
<code>SrcObjectNotEligible</code>	Das Objekt kann nicht repliziert werden. Das Objekt oder seine Objekt-Tags stimmt/stimmen nicht mit der Replikationskonfiguration überein.

Weitere Informationen zur Behebung von Replikationsfehlern finden Sie in folgenden Themen:

- [Erstellen einer IAM-Rolle](#)
- [Fehlerbehebung bei einer Replikation](#)

Überwachen von EventBridge mit CloudWatch

Für die Überwachung lässt sich Amazon EventBridge mit Amazon CloudWatch integrieren. EventBridge sendet automatisch jede Minute Metriken an CloudWatch. Zu diesen Metriken gehören die Anzahl der [Ereignisse](#), die von einer [Regel](#) abgeglichen wurden, sowie die Anzahl der Aufrufe eines [Ziels](#) durch eine Regel. Wenn eine Regel in EventBridge ausgeführt wird, werden alle Ziele aufgerufen, die mit der Regel verknüpft sind. Sie können das Verhalten von EventBridge wie folgt mit CloudWatch überwachen.

- Sie können die verfügbaren [EventBridge-Metriken](#) für Ihre EventBridge-Regeln über das CloudWatch-Dashboard überwachen. Anschließend können Sie CloudWatch-Funktionen wie beispielsweise CloudWatch-Alarme verwenden, um Alarme für bestimmte Metriken einzustellen. Wenn diese Metriken die benutzerdefinierten Schwellenwerte erreichen, die Sie in den Alarmen angegeben haben, erhalten Sie Benachrichtigungen und können entsprechende Maßnahmen ergreifen.
- Sie können Amazon CloudWatch Logs als Ziel Ihrer EventBridge-Regel festlegen. EventBridge erstellt dann Protokollstreams und CloudWatch Logs speichert den Text der Ereignisse als Protokolleinträge. Weitere Informationen finden Sie unter [EventBridge und CloudWatch Logs](#).

Weitere Informationen zum Debuggen von EventBridge-Ereignisübermittlungs- und -archivierungsereignissen finden Sie unter den folgenden Themen:

- [Richtlinie zur Wiederholung von Ereignissen und Verwendung von Warteschlangen für unzustellbare Nachrichten](#)
- [Archivieren von EventBridge-Ereignissen](#)

S3 auf Outposts teilen mit AWS RAM

Amazon S3 on Outposts unterstützt die gemeinsame Nutzung von S3-Kapazität für mehrere Konten innerhalb einer Organisation mithilfe von AWS Resource Access Manager ([AWS RAM](#)). Mit der Freigabe von S3 in Outposts können Sie anderen erlauben, Buckets, Endpunkte und Zugriffspunkte in Ihrem Outpost zu erstellen und zu verwalten.

In diesem Thema wird gezeigt, wie Sie AWS RAM S3 auf Outposts und verwandte Ressourcen mit anderen AWS-Konto in Ihrer AWS Organisation teilen können.

Voraussetzungen

- Für das Outpost-Eigentümerkonto ist eine Organisation in AWS Organizations konfiguriert. Weitere Informationen finden Sie unter [Erstellen einer Organisation](#) im Benutzerhandbuch für AWS Organizations .
- Die Organisation umfasst AWS-Konto die, mit der Sie Ihre S3 on Outposts-Kapazität teilen möchten. Weitere Informationen finden Sie unter [Senden von Einladungen an AWS-Konten](#) im Benutzerhandbuch für AWS Organizations .
- Wählen Sie eine der folgenden Optionen, die Sie freigeben möchten. Die zweite Ressource (entweder Subnets (Subnetze) oder Outposts) muss ausgewählt sein, damit auch Endpunkte zugänglich sind. Endpunkte sind eine Netzwerkanforderung, um auf Daten zuzugreifen, die in S3 in Outposts gespeichert sind.

Option 1	Option 2
S3 in Outposts	S3 in Outposts
Erlaubt es dem Benutzer, Buckets auf Ihren Outposts und Zugriffspunkten zu erstellen und diesen Buckets Objekte hinzuzufügen.	Erlaubt es dem Benutzer, Buckets auf Ihren Outposts und Zugriffspunkten zu erstellen und diesen Buckets Objekte hinzuzufügen.
Subnets	Outposts
Ermöglicht dem Benutzer die Nutzung Ihrer virtuellen privaten Cloud (VPC) und der Endpunkte, die Ihrem Subnetz zugeordnet sind.	Erlaubt dem Benutzer das Anzeigen von S3-Kapazitätstabellen und der AWS Outposts -Konsolen-Startseite. Erlaubt es Benutzern außerdem, Subnetze auf freigegebenen Outposts zu erstellen und Endpunkte zu erstellen.

Verfahren

1. [Melden Sie sich mit AWS Management Console dem an AWS-Konto , dem der Outpost gehört, und öffnen Sie dann die AWS RAM Konsole unter <https://console.aws.amazon.com/ram>.](#)

2. Vergewissern Sie sich, dass Sie das Teilen mit AWS Organizations in AWS RAM aktiviert haben. Weitere Informationen finden Sie unter [Freigabe für Ressourcen in AWS Organizations aktivieren](#) im AWS RAM -Benutzerhandbuch.
3. Verwenden Sie entweder Option 1 oder Option 2 in den [Voraussetzungen](#), um eine Ressourcenfreigabe zu erstellen. Wenn Sie über mehrere S3 on Outposts-Ressourcen verfügen, wählen Sie die Amazon-Ressourcennamen (ARNs) der Ressourcen aus, die Sie teilen möchten. Wenn Sie Endpunkte aktivieren möchten, teilen Sie entweder Ihr Subnetz oder Ihren Outpost.

Weitere Informationen zum Erstellen einer Ressourcenfreigabe finden Sie unter [Erstellen einer Ressourcenfreigabe](#) im AWS RAM -Benutzerhandbuch.

4. Die AWS-Konto Personen, mit denen Sie Ihre Ressourcen geteilt haben, sollten jetzt S3 auf Outposts verwenden können. Abhängig von der Option, die Sie in den [Voraussetzungen](#) gewählt haben, geben Sie dem Kontobenutzer die folgenden Informationen an:

Option 1	Option 2
Die Outpost-ID	Die Outpost-ID
Die VPC ID	
Die Subnetz-ID	
Die Sicherheitsgruppen-ID	

Note

Der Benutzer kann mithilfe der AWS RAM Konsole AWS Command Line Interface (AWS CLI), oder bestätigen, dass die Ressourcen für ihn freigegeben wurden RESTAPI. AWS SDKs Der Benutzer kann seine vorhandenen Ressourcenfreigaben mithilfe des [get-resource-shares](#)CLIBefehls einsehen.

Verwendungsbeispiele

Nachdem Sie Ihre S3 in Outposts-Ressourcen mit einem anderen Konto geteilt haben, kann dieses Konto Buckets und Objekte in Ihrem Outpost verwalten. Wenn Sie die Ressource Subnets (Subnetze) freigegeben haben, kann dieses Konto den von Ihnen erstellten Endpunkt verwenden. Die folgenden

Beispiele zeigen, wie ein Benutzer den verwenden kann AWS CLI , um mit Ihrem Outpost zu interagieren, nachdem Sie diese Ressourcen gemeinsam genutzt haben.

Example : Erstellen eines Buckets

Im folgenden Beispiel wird ein Bucket mit dem Namen erstellt *amzn-s3-demo-bucket1* auf dem Outpost *op-01ac5d28a6a232904*. Bevor Sie diesen Befehl verwenden, ersetzen Sie jeden *user input placeholder* mit den entsprechenden Werten für Ihren Anwendungsfall.

```
aws s3control create-bucket --bucket amzn-s3-demo-bucket1 --outpost-id op-01ac5d28a6a232904
```

Weitere Informationen über diesen Befehl finden Sie unter [create-bucket](#) in der AWS CLI -Referenz.

Example : Erstellen eines Zugriffspunkts

Im folgenden Beispiel wird ein Zugriffspunkt in einem Outpost erstellt, wobei die Beispielparameter in der folgenden Tabelle verwendet werden. Bevor Sie diesen Befehl verwenden, ersetzen Sie diese *user input placeholder* Werte und den AWS-Region Code durch die entsprechenden Werte für Ihren Anwendungsfall.

Parameter	Wert
Konto-ID	<i>111122223333</i>
Name des Zugriffspunkts	<i>example-outpost-access-point</i>
Outpost-ID	<i>op-01ac5d28a6a232904</i>
Name des Outpost-Buckets	<i>amzn-s3-demo-bucket1</i>
VPCID	<i>vpc-1a2b3c4d5e6f7g8h9</i>

Note

Der Account-ID-Parameter muss die ID des Bucket-Besitzers sein, bei dem es sich um den gemeinsamen Benutzer handelt. AWS-Konto

```
aws s3control create-access-point --account-id 111122223333 --name example-outpost-  
access-point \  
--bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/  
bucket/amzn-s3-demo-bucket1 \  
--vpc-configuration VpcId=vpc-1a2b3c4d5e6f7g8h9
```

Weitere Informationen zu diesem Befehl finden Sie [create-access-point](#) in der AWS CLI Referenz.

Example : Hochladen eines Objekts

Im folgenden Beispiel wird die Datei hochgeladen *my_image.jpg* vom lokalen Dateisystem des Benutzers zu einem Objekt mit dem Namen *images/my_image.jpg* durch den Access Point *example-outpost-access-point* auf dem Außenposten *op-01ac5d28a6a232904*, im Besitz des Accounts AWS *111122223333*. Bevor Sie diesen Befehl verwenden, ersetzen Sie diese *user input placeholder* Werte und den AWS-Region Code durch die entsprechenden Werte für Ihren Anwendungsfall.

```
aws s3api put-object --bucket arn:aws:s3-outposts:us-  
east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-  
point \  
--body my_image.jpg --key images/my_image.jpg
```

Weitere Informationen über diesen Befehl finden Sie unter [put-object](#) in der AWS CLI -Referenz.

Note

Wenn dieser Vorgang zu einem Fehler „Ressource nicht gefunden“ führt oder nicht reagiert, haben Sie VPC möglicherweise keinen gemeinsamen Endpunkt.

Verwenden Sie den [list-shared-endpoints](#) AWS CLI Befehl, um zu überprüfen, ob es einen gemeinsamen Endpunkt gibt. Wenn kein freigegebener Endpunkt vorhanden ist, erstellen Sie einen Endpunkt gemeinsam mit dem Outpost-Besitzer. Weitere Informationen finden Sie [ListSharedEndpoints](#) in der Amazon Simple Storage Service API Reference.

Example : Erstellen eines Endpunkts

Das folgende Beispiel erstellt einen Endpunkt für einen freigegebenen Outpost. Bevor Sie diesen Befehl verwenden, ersetzen Sie die *user input placeholder*-Werte für die Outpost-ID, die Subnetz-ID und die Sicherheitsgruppen-ID durch die entsprechenden Werte für Ihren Anwendungsfall.

Note

Der Benutzer kann diesen Vorgang nur ausführen, wenn die Ressourcenfreigabe die Outposts-Ressource enthält.

```
aws s3outposts create-endpoint --outposts-id op-01ac5d28a6a232904 --subnet-id XXXXXX --
security-group-id XXXXXXXX
```

Weitere Informationen über diesen Befehl finden Sie unter [create-endpoint](#) in der AWS CLI - Referenz.

Sonstige AWS-Services, die S3 on Outposts verwenden

Andere AWS-Services, die lokal auf Ihrem AWS Outposts ausgeführt werden, können Ihre Kapazität von Amazon S3 on Outposts ebenfalls verwenden. In Amazon CloudWatch zeigt der Namespace `S3Outposts` detaillierte Metriken für Buckets in S3 on Outposts an. Diese Metriken berücksichtigen jedoch keine Nutzung anderer AWS-Services. Informationen zum Verwalten Ihrer S3-on-Outposts-Kapazität, die von anderen AWS-Services verbraucht wird, finden Sie in der folgenden Tabelle.

AWS-Service	Beschreibung	Weitere Informationen
Amazon S3	Die gesamte direkte S3-on-Outposts-Nutzung verfügt über eine CloudWatch-Metrik zum übereinstimmenden Konto und Bucket.	Siehe Metriken
Amazon Elastic Block Store (Amazon EBS)	Für Amazon EBS on Outposts können Sie einen AWS-Outpost als Snapshot-Ziel auswählen und in Ihrem S3 on Outpost lokal speichern.	Weitere Informationen
Amazon Relational Database Service (Amazon RDS)	Sie können lokale Amazon-RDS-Backups verwenden, um Ihre RDS-Backups lokal in Ihrem Outpost zu speichern.	Weitere Informationen

Überwachen von S3 in Outposts

Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihren AWS Outposts erstellen und Objekte für Anwendungen, die lokalen Datenzugriff, lokale Datenverarbeitung und Datenresidenz erfordern, einfach vor Ort speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse, S3 Outposts (OUTPOSTS), die Amazon S3 verwendet und darauf ausgelegt ist APIs, Daten dauerhaft und redundant auf mehreren Geräten und Servern auf Ihrem zu speichern. AWS Outposts Sie kommunizieren mit Ihrem Outpost-Bucket, indem Sie einen Access Point und eine Endpunktverbindung über eine virtuelle private Cloud (VPC) verwenden. Sie können für Outpost-Buckets dieselben APIs Funktionen wie für Amazon S3 S3-Buckets verwenden, einschließlich Zugriffsrichtlinien, Verschlüsselung und Tagging. Sie können S3 auf Outposts über das AWS Management Console, AWS Command Line Interface (AWS CLI) AWS SDKs, oder REST API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Weitere Informationen zum Überwachen Ihrer Speicherkapazität von Amazon S3 in Outposts finden Sie in den folgenden Themen.

Themen

- [Verwaltung der Kapazität von S3 on Outposts mit Amazon-Metriken CloudWatch](#)
- [Empfangen von S3 on Outposts-Ereignisbenachrichtigungen mithilfe von Amazon CloudWatch Events](#)
- [Überwachen von S3 in Outposts mit Protokollen in AWS CloudTrail](#)

Verwaltung der Kapazität von S3 on Outposts mit Amazon-Metriken CloudWatch

Um Ihnen die Verwaltung der festen S3-Kapazität in Ihrem Outpost zu erleichtern, empfehlen wir Ihnen, CloudWatch Warnmeldungen zu erstellen, die Sie darüber informieren, wenn Ihre Speichernutzung einen bestimmten Schwellenwert überschreitet. Weitere Informationen zu den CloudWatch Metriken für S3 auf Outposts finden Sie unter [CloudWatch Metriken](#). Wenn in Ihrem Outpost nicht genügend Speicherplatz zum Speichern eines Objekts vorhanden ist, wird eine Ausnahme für unzureichende Kapazität API zurückgegeben (ICE). Um Speicherplatz freizugeben, können Sie CloudWatch Alarmer erstellen, die eine explizite Datenlöschung auslösen, oder eine Lebenszyklus-Ablaufrichtlinie verwenden, um Objekte ablaufen zu lassen. Um Daten vor dem Löschen zu speichern, können AWS DataSync Sie Daten aus Ihrem Amazon S3 on Outposts

Bucket in einen S3-Bucket in einem AWS-Region kopieren. Weitere Informationen zur Verwendung DataSync finden Sie unter [Erste Schritte mit AWS DataSync](#) im AWS DataSync Benutzerhandbuch.

CloudWatch Metriken

Der S3Outposts Namespace enthält die folgenden Metriken für Amazon S3 auf Outposts-Buckets. Sie können die Gesamtzahl der bereitgestellten S3 in Outposts-Bytes, die für Objekte insgesamt verfügbaren freien Bytes und die Gesamtgröße aller Objekte für einen bestimmten Bucket überwachen. Bucket- oder kontobezogene Metriken gibt es für die gesamte direkte S3-Nutzung. Die indirekte S3-Nutzung, wie das Speichern lokaler Snapshots von Amazon Elastic Block Store oder Backups von Amazon Relational Database Service auf einem Outpost, verbraucht S3-Kapazität, ist aber nicht in den Bucket- oder kontobezogenen Metriken enthalten. Weitere Informationen zu EBS lokalen Amazon-Snapshots finden Sie unter [EBS Lokale Amazon-Snapshots auf Outposts](#). Ihren EBS Amazon-Kostenbericht finden Sie unter <https://console.aws.amazon.com/billing/>.

Note

S3 in Outposts unterstützt nur die folgenden Metriken und keine anderen Amazon-S3-Metriken.

Da S3 on Outposts ein festes Kapazitätslimit hat, empfehlen wir, CloudWatch Alarmer zu erstellen, um Sie zu benachrichtigen, wenn Ihre Speichernutzung einen bestimmten Schwellenwert überschreitet.

Metrik	Beschreibung	Zeitraum	Einheiten	Typ
OutpostTotalBytes	Die gesamte bereitgestellte Kapazität in Byte für einen Outpost	5 Minuten	Bytes	S3 on Outposts
OutpostFreeBytes	Die Anzahl der freien Bytes, die auf Outposts zum Speichern von Kundendaten verfügbar sind.	5 Minuten	Bytes	S3 on Outposts
BucketUsedBytes	Die Gesamtgröße aller Objekte für den angegebenen Bucket.	5 Minuten	Bytes	S3 on Outposts Nur direkte S3-Nutzung

Metrik	Beschreibung	Zeitraum	Einheiten	Typ
AccountTotalBytes	Die Gesamtgröße aller Objekte für das angegebene Outposts-Konto.	5 Minuten	Bytes	S3 on Outposts Nur direkte S3-Nutzung
BytesPerReplication	Die Gesamtanzahl der Bytes von Objekten, deren Replikation für eine bestimmte Replikationsregel aussteht. Weitere Informationen zum Aktivieren von Replikationsmetriken finden Sie unter Erstellen von Replikationsregeln zwischen Outposts .	5 Minuten	Bytes	Optional. Für S3 Replication in Outposts.
OperationsPending	Die Gesamtanzahl der Operationen, deren Replikation für eine bestimmte Replikationsregel aussteht. Weitere Informationen zum Aktivieren von Replikationsmetriken finden Sie unter Erstellen von Replikationsregeln zwischen Outposts .	5 Minuten	Zählungen	Optional. Für S3 Replication in Outposts.
ReplicationLatency	Die aktuelle Verzögerung in Sekunden, um die der Replikationsziel-Bucket hinter dem Quell-Bucket für eine bestimmte Replikationsregel zurückliegt. Weitere Informationen zum Aktivieren von Replikationsmetriken finden Sie unter Erstellen von Replikationsregeln zwischen Outposts .	5 Minuten	Sekunden	Optional. Für S3 Replication in Outposts.

Empfangen von S3 on Outposts-Ereignisbenachrichtigungen mithilfe von Amazon CloudWatch Events

Sie können CloudWatch Ereignisse verwenden, um eine Regel für jedes Amazon S3 on API Outposts-Ereignis zu erstellen. Wenn Sie eine Regel erstellen, können Sie wählen, ob Sie über alle unterstützten CloudWatch Ziele benachrichtigt werden möchten, einschließlich Amazon Simple Queue Service (AmazonSQS), Amazon Simple Notification Service (AmazonSNS) und AWS Lambda. Weitere Informationen finden Sie in der Liste der [AWS Dienste, die Ziele für CloudWatch Veranstaltungen sein können](#), im Amazon CloudWatch Events-Benutzerhandbuch. Informationen zur Auswahl eines Zieldienstes, der mit Ihrem S3 auf Outposts funktioniert, finden Sie im Amazon CloudWatch CloudWatch Events-Benutzerhandbuch [unter Erstellen einer Ereignisregel, die AWS CloudTrail bei einem AWS API Aufruf auslöst](#) wird.

Note

Bei Objektoperationen in S3 on Outposts entsprechen die von gesendeten AWS API Aufrufereignisse nur CloudTrail dann Ihren Regeln, wenn Sie Trails (optional mit Event-Selektoren) für den Empfang dieser Ereignisse konfiguriert haben. Weitere Informationen finden Sie unter [Arbeiten mit CloudTrail Protokolldateien](#) im AWS CloudTrail Benutzerhandbuch.

Example

Es folgt eine Beispielregel für den DeleteObject-Vorgang. Zum Verwenden dieser Beispielregel ersetzen Sie *amzn-s3-demo-bucket1* durch den Namen Ihres S3-in-Outposts-Buckets.

```
{
  "source": [
    "aws.s3-outposts"
  ],
  "detail-type": [
    "AWS API call through CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3-outposts.amazonaws.com"
    ],
    "eventName": [
      "DeleteObject"
    ]
  }
}
```

```
    ],  
    "requestParameters": {  
      "bucketName": [  
        "amzn-s3-demo-bucket1"  
      ]  
    }  
  }  
}
```

Überwachen von S3 in Outposts mit Protokollen in AWS CloudTrail

Amazon S3 on Outposts ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS-Service in S3 auf Outposts ausgeführten Aktionen bereitstellt. Sie können mit AWS CloudTrail Informationen über Anforderungen auf Bucket- und Objektebene in S3 in Outposts abrufen, um Ihre S3-in-Outposts-Ereignisaktivitäten zu überprüfen und zu protokollieren. Um CloudTrail Datenereignisse für alle Ihre Outposts-Buckets oder für eine Liste bestimmter Outposts-Buckets zu aktivieren, müssen Sie manuell [einen Trail in erstellen](#). CloudTrail Weitere Informationen zu CloudTrail Protokolldateieinträgen finden Sie unter [Protokolldateieinträge von S3 on Outposts](#).

Note

- Es hat sich bewährt, eine Lebenszyklusrichtlinie für Ihren Outposts-Bucket für AWS CloudTrail Datenereignisse zu erstellen. Konfigurieren Sie die Lebenszyklusrichtlinie zum regelmäßigen Entfernen von Protokolldateien nach dem Zeitraum, der für die Überprüfung erforderlich ist. Dadurch wird die Menge der Daten reduziert, die Amazon Athena in einer Abfrage analysiert. Weitere Informationen finden Sie unter [Einstellung einer Lebenszykluskonfiguration für einen Bucket](#).
- Beispiele für das Abfragen von CloudTrail Protokollen finden Sie im AWS Big-Data-Blogbeitrag [Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail and Amazon Athena](#).

Aktivieren CloudTrail Sie die Protokollierung für Objekte in einem S3-Bucket auf Outposts


Sie können die Amazon S3-Konsole verwenden, um einen AWS CloudTrail Trail zu konfigurieren, um Datenereignisse für Objekte in einem Amazon S3 on Outposts-Bucket zu protokollieren. CloudTrail

unterstützt die Protokollierung von S3 Outposts API Outposts-Operationen auf Objektebene wie `GetObject`, `deleteObject`, `PutObject`. Diese Ereignisse werden als Datenereignisse bezeichnet.

Standardmäßig protokollieren CloudTrail Trails keine Datenereignisse. Sie können Trails jedoch so konfigurieren, dass sie Datenereignisse für von Ihnen festgelegte S3-in-Outposts-Buckets protokollieren oder dass sie Datenereignisse für alle S3-in-Outposts-Buckets in Ihrem AWS-Konto protokollieren. Weitere Informationen finden Sie unter [Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail](#).


CloudTrail fügt keine Datenereignisse in den CloudTrail Ereignisverlauf ein. Darüber hinaus sind nicht alle API Operationen auf Bucket-Ebene von S3 on Outposts in der Ereignishistorie aufgeführt. CloudTrail Weitere Informationen zum Abfragen von CloudTrail Protokollen finden Sie im AWS Knowledge Center [unter Verwenden von Amazon CloudWatch Logs-Filtermustern und Amazon Athena zum Abfragen von CloudTrail Protokollen](#).

Um einen Trail zum Protokollieren von Datenereignissen für einen S3-in-Outposts-Bucket zu konfigurieren, können Sie entweder die AWS CloudTrail -Konsole oder die Amazon-S3-Konsole verwenden. Wenn Sie einen Trail konfigurieren, um Datenereignisse für alle S3 on Outposts-Buckets in Ihrem zu protokollieren AWS-Konto, ist es einfacher, die CloudTrail Konsole zu verwenden. Informationen zur Verwendung der CloudTrail Konsole zur Konfiguration eines Trails zur Protokollierung von Datenereignissen von S3 on Outposts finden Sie unter [Datenereignisse](#) im AWS CloudTrail Benutzerhandbuch.

 **Important**

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen finden Sie unter [AWS CloudTrail – Preise](#).

Das folgende Verfahren zeigt, wie Sie mit der Amazon S3 S3-Konsole einen CloudTrail Trail zum Protokollieren von Datenereignissen für einen S3 on Outposts-Bucket konfigurieren.

 **Note**

Derjenige, der den Bucket erstellt AWS-Konto , besitzt ihn und ist der einzige, der S3 on Outposts konfigurieren kann, an die Datenereignisse gesendet werden sollen. AWS CloudTrail

Um die Protokollierung CloudTrail von Datenereignissen für Objekte in einem S3 on Outposts-Bucket zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Navigationsbereich Outposts buckets (Outposts-Buckets) aus.
3. Wählen Sie den Namen des Outposts-Buckets, mit CloudTrail dessen Datenereignissen Sie protokollieren möchten.
4. Wählen Sie Properties (Eigenschaften).
5. Wählen Sie im Bereich AWS CloudTrail Datenereignisse die Option Konfigurieren in CloudTrail aus.

Die AWS CloudTrail Konsole wird geöffnet.

Sie können einen neuen CloudTrail Trail erstellen oder einen vorhandenen Trail wiederverwenden und S3 on Outposts Datenereignisse so konfigurieren, dass sie in Ihrem Trail protokolliert werden.

6. Wählen Sie auf der Dashboard-Seite der CloudTrail Konsole die Option Create Trail aus.
7. Geben Sie auf der Seite Schritt 1 Trail-Attribute auswählen einen Namen für den Trail ein, wählen Sie einen S3-Bucket als Speicherort für die Trail-Protokolle aus, geben Sie alle weiteren gewünschten Einstellungen an und wählen Sie dann Nächstes aus.
8. Wählen Sie auf der Seite Schritt 2 Protokollereignisse auswählen unter Ereignistyp die Option Datenereignisse aus.

Wählen Sie als Datenereignistyp S3 Outposts aus. Wählen Sie Weiter aus.

Note

- Wenn Sie einen Trail erstellen und die Datenereignisprotokollierung für S3 in Outposts konfigurieren, müssen Sie den Datenereignistyp korrekt angeben.
- Wenn Sie die CloudTrail Konsole verwenden, wählen Sie S3 Outposts als Datenereignistyp. Informationen zum Erstellen von Trails in der CloudTrail Konsole finden Sie im AWS CloudTrail Benutzerhandbuch unter [Erstellen und Aktualisieren eines Trails mit der Konsole](#). Informationen zur Konfiguration der Datenereignisprotokollierung von S3 on Outposts in der CloudTrail Konsole finden

Sie unter [Protokollierung von Datenereignissen für Amazon S3 S3-Objekte](#) im AWS CloudTrail Benutzerhandbuch.

- Wenn Sie das AWS Command Line Interface (AWS CLI) oder das verwenden AWS SDKs, setzen Sie das `resources.type` Feld auf `AWS::S3Outposts::Object`. Weitere Informationen zum Protokollieren von S3 on Outposts-Datenereignissen mit dem AWS CLI finden Sie unter [S3 on Outposts-Ereignisse protokollieren](#) im AWS CloudTrail Benutzerhandbuch.
- Wenn Sie die CloudTrail Konsole oder die Amazon S3 S3-Konsole verwenden, um einen Trail zur Protokollierung von Datenereignissen für einen S3 on Outposts-Bucket zu konfigurieren, zeigt die Amazon S3 S3-Konsole an, dass die Protokollierung auf Objektebene für den Bucket aktiviert ist.

9. Überprüfen Sie auf der Seite Schritt 3 Überprüfen und erstellen die von Ihnen konfigurierten Trail-Attribute und Protokollereignisse. Wählen Sie dann Trail erstellen aus.

So deaktivieren Sie die Protokollierung von CloudTrail Datenereignissen für Objekte in einem S3-Bucket auf Outposts

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die CloudTrail Konsole unter <https://console.aws.amazon.com/cloudtrail/>.
2. Wählen Sie im linken Navigationsbereich Trails aus.
3. Wählen Sie den Namen des Trails aus, den Sie erstellt haben, um Ereignisse für den S3-in-Outposts-Bucket zu protokollieren.
4. Wählen Sie oben rechts auf der Detailseite des Trails Protokollierung beenden aus.
5. Wählen Sie im anschließend angezeigten Dialogfeld Protokollierung beenden aus.

Entwickeln mit Amazon S3 on Outposts

Mit Amazon S3 in Outposts können Sie S3-Buckets auf Ihren AWS-Outposts erstellen und Objekte für Anwendungen, die einen lokalen Datenzugriff, eine lokale Datenverarbeitung und eine lokale Datenresidenz erfordern, auf einfache Weise On-Premise speichern und abrufen. S3 on Outposts bietet eine neue Speicherklasse namens S3 Outposts (OUTPOSTS). Sie nutzt die Amazon-S3-APIs und ist darauf ausgelegt, Daten über mehrere Geräte und Server hinweg in Ihrem AWS Outposts dauerhaft und redundant zu speichern. Sie kommunizieren mit Ihrem Outpost-Bucket über einen Zugriffspunkt und eine Endpunktverbindung über eine Virtual Private Cloud (VPC). Sie können bei

Outpost-Buckets dieselben APIs und Funktionen wie in Amazon-S3-Buckets verwenden, inklusive Zugriffsrichtlinien, Verschlüsselung und Markierungen. Sie können S3 on Outposts über die AWS Management Console, AWS Command Line Interface (AWS CLI), AWS-SDKs oder die REST-API verwenden. Weitere Informationen finden Sie unter [Was ist Amazon S3 on Outposts?](#).

Die folgenden Themen enthalten Informationen zur Entwicklung mit S3 on Outposts.

Themen

- [API-Vorgänge in Amazon S3 on Outposts](#)
- [Konfigurieren des S3-Steuerungsclients für S3 on Outposts mit dem SDK for Java](#)
- [Senden von Anforderungen an S3 on Outposts über IPv6](#)

API-Vorgänge in Amazon S3 on Outposts

In diesem Thema werden die API-Vorgänge für Amazon S3, Amazon S3 Control und Amazon S3 on Outposts aufgeführt, die Sie mit Amazon S3 on Outposts verwenden können.

Themen

- [Amazon-S3-API-Vorgänge für die Objektverwaltung](#)
- [Amazon-S3-Control-API-Vorgänge zum Verwalten von Buckets](#)
- [S3-on-Outposts-API-Vorgänge zur Verwaltung von Outposts](#)

Amazon-S3-API-Vorgänge für die Objektverwaltung

S3 on Outposts ist so konzipiert, dass es die gleichen Objekt-API-Vorgänge wie Amazon S3 verwendet. Sie müssen Zugriffspunkte verwenden, um auf ein Objekt in einem Outpost-Bucket zuzugreifen. Wenn Sie eine Objekt-API-Operation mit S3 on Outposts verwenden, geben Sie entweder den Amazon-Ressourcennamen (ARN) des Zugriffspunkts für Outposts oder den Zugriffspunkt-Alias an. Weitere Informationen zu Zugriffspunkt-Aliassen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

Amazon S3 on Outposts unterstützt die folgenden Amazon-S3-API-Operationen:

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CopyObject](#)

- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [DeleteObjectTagging](#)
- [GetObject](#)
- [GetObjectTagging](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjects](#)
- [ListObjectsV2](#)
- [ListObjectVersions](#)
- [ListParts](#)
- [PutObject](#)
- [PutObjectTagging](#)
- [UploadPart](#)
- [UploadPartCopy](#)

Amazon-S3-Control-API-Vorgänge zum Verwalten von Buckets

S3 on Outposts unterstützt die folgenden Amazon-S3-Control-API-Vorgänge für die Arbeit mit Buckets.

- [CreateAccessPoint](#)
- [CreateBucket](#)
- [DeleteAccessPoint](#)
- [DeleteAccessPointPolicy](#)
- [DeleteBucket](#)
- [DeleteBucketLifecycleConfiguration](#)
- [DeleteBucketPolicy](#)
- [DeleteBucketReplication](#)
- [DeleteBucketTagging](#)

- [GetAccessPoint](#)
- [GetAccessPointPolicy](#)
- [GetBucket](#)
- [GetBucketLifecycleConfiguration](#)
- [GetBucketPolicy](#)
- [GetBucketReplication](#)
- [GetBucketTagging](#)
- [GetBucketVersioning](#)
- [ListAccessPoints](#)
- [ListRegionalBuckets](#)
- [PutAccessPointPolicy](#)
- [PutBucketLifecycleConfiguration](#)
- [PutBucketPolicy](#)
- [PutBucketReplication](#)
- [PutBucketTagging](#)
- [PutBucketVersioning](#)

S3-on-Outposts-API-Vorgänge zur Verwaltung von Outposts

S3 on Outposts unterstützt die folgenden API-Vorgänge für Amazon S3 on Outposts zur Verwaltung von Endpunkten.

- [CreateEndpoint](#)
- [DeleteEndpoint](#)
- [ListEndpoints](#)
- [ListOutpostsWithS3](#)
- [ListSharedEndpoints](#)

Konfigurieren des S3-Steuerungsclients für S3 on Outposts mit dem SDK for Java

Im folgenden Beispiel wird der Amazon-S3-Steuerungs-Client für Amazon S3 on Outposts mithilfe von AWS SDK for Java konfiguriert. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

```
import com.amazonaws.auth.AWSSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;

public AWSS3Control createS3ControlClient() {

    String accessKey = AWSAccessKey;
    String secretKey = SecretAccessKey;
    BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);

    return AWSS3ControlClient.builder().enableUseArnRegion()
        .withCredentials(new AWSSStaticCredentialsProvider(awsCreds))
        .build();
}
```

Senden von Anforderungen an S3 on Outposts über IPv6

Dual-Stack-Endpunkte von Amazon S3 on Outposts und S3 on Outposts unterstützen Anfragen an S3-on-Outposts-Buckets entweder über das IPv6- oder IPv4-Protokoll. Mit IPv6-Unterstützung für S3 on Outposts können Sie über S3-on-Outposts-APIs über IPv6-Netzwerke auf Ihre Buckets und Stueberebenenressourcen zugreifen und diese betreiben.

Note

[Objektaktionen von S3 on Outposts](#) (wie PutObject oder GetObject) werden über IPv6-Netzwerke nicht unterstützt.

Für den Zugriff auf S3 on Outposts über IPv6-Netzwerke fallen keine zusätzlichen Gebühren an. Weitere Informationen zu S3 on Outposts finden Sie unter [S3 on Outposts – Preise](#).

Themen

- [Erste Schritte mit IPv6](#)
- [Verwenden von Dual-Stack-Endpunkten für Anfragen über ein IPv6-Netzwerk](#)
- [Verwenden von IPv6-Adressen in IAM-Richtlinien](#)
- [Testen der IP-Adresskompatibilität](#)
- [Verwenden von IPv6 mit AWS PrivateLink](#)
- [Verwenden von S3-on-Outposts-Dual-Stack-Endpunkten](#)

Erste Schritte mit IPv6

Um eine Anforderung über IPv6 an einen S3-on-Outposts-Bucket zu stellen, müssen Sie einen Dual-Stack-Endpunkt verwenden. Der nächste Abschnitt beschreibt Anfragen über IPv6 unter Verwendung von Dual-Stack-Endpunkten.

Die folgenden wichtigen Überlegungen sind zu beachten, bevor Sie versuchen, über IPv6 auf einen S3-on-Outposts-Bucket zuzugreifen:

- Der Client und das Netzwerk, die auf den Bucket zugreifen, müssen für IPv6 aktiviert sein.
- Für den IPv6-Zugriff werden Anforderungen im virtuellen Hosting- und im Pfad-Stil unterstützt. Weitere Informationen finden Sie unter [Verwenden von S3-on-Outposts-Dual-Stack-Endpunkten](#).
- Wenn Sie die Quell-IP-Adressfilterung in Ihren AWS Identity and Access Management (IAM)-Benutzer- oder S3-on-Outposts-Bucket-Richtlinien verwenden, müssen Sie die Richtlinien so aktualisieren, dass sie IPv6-Adressbereiche enthalten.

Note

Diese Anforderung gilt nur für S3-on-Outposts-Bucket-Operationen und Ressourcen der Steuerebene in allen IPv6-Netzwerken. [Objektaktionen von Amazon S3 on Outposts](#) werden in IPv6-Netzwerken nicht unterstützt.

- Bei Verwendung von IPv6 geben die Serverzugriff-Protokolldateien IP-Adressen in einem IPv6-Format aus. Sie müssen vorhandene Tools, Skripts und Software aktualisieren, die Sie zum Analysieren von S3-on-Outposts-Protokolldateien verwenden, damit sie die IPv6-formatierten Remote-IP-Adressen analysieren können. Die aktualisierten Tools, Skripts und Software analysieren dann die IPv6-formatierten Remote-IP-Adressen korrekt.

Verwenden von Dual-Stack-Endpunkten für Anfragen über ein IPv6-Netzwerk

Um Anforderungen mit S3-on-Outposts-API-Aufrufen über IPv6 zu stellen, können Sie Dual-Stack-Endpunkte über AWS CLI oder AWS SDK verwenden. Die [Amazon S3-Steuerungs-API-Operationen](#) und [S3-on-Outposts-API-Operationen](#) funktionieren auf die gleiche Weise, unabhängig davon, ob Sie über ein IPv6-Protokoll oder ein IPv4-Protokoll auf S3 on Outposts zugreifen. Beachten Sie jedoch, dass [Objektaktionen von S3 on Outposts](#) (z. B. PutObject oder GetObject) über IPv6-Netzwerke nicht unterstützt werden.

Wenn Sie die AWS Command Line Interface (AWS CLI) und AWS SDKs verwenden, können Sie einen Parameter oder ein Flag verwenden, um zu einem Dual-Stack-Endpunkt zu wechseln. Sie können den Dual-Stack-Endpunkt auch direkt als Überschreibung des S3-on-Outposts-Endpunkts in der Konfigurationsdatei angeben.

Sie können einen Dual-Stack-Endpunkt verwenden, um über IPv6 von einer der folgenden Optionen auf einen S3-on-Outposts-Bucket zuzugreifen:

- Die AWS CLI; siehe [Verwenden von Dual-Stack-Endpunkten von der AWS CLI](#).
- Die AWS-SDKs finden Sie unter [Verwenden von S3-on-Outposts-Dual-Stack-Endpunkten über die AWS -SDKs](#).

Verwenden von IPv6-Adressen in IAM-Richtlinien

Bevor Sie versuchen, mit einem IPv6-Protokoll auf einen S3-on-Outposts-Bucket zuzugreifen, stellen Sie sicher, dass IAM-Benutzer oder S3-on-Outposts-Bucket-Richtlinien, die für die IP-Adressfilterung verwendet werden, aktualisiert werden, um IPv6-Adressbereiche einzuschließen. Wenn IP-Adressenfilterrichtlinien nicht für die Verarbeitung von IPv6-Adressen aktualisiert werden, können Sie den Zugriff auf einen S3-on-Outposts-Bucket verlieren, während Sie versuchen, das IPv6-Protokoll zu verwenden.

IAM-Richtlinien, die IP-Adressen filtern, verwenden [Bedingungsoperatoren für IP-Adressen](#). Die folgende Bucket-Richtlinie für S3 on Outposts identifiziert die 54.240.143.* IP-Bereich zulässiger IPv4-Adressen unter Verwendung von Bedingungsoperatoren für IP-Adressen. Alle IP-Adressen außerhalb dieses Bereichs erhalten keinen Zugriff auf den S3-on-Outposts-Bucket (DOC-EXAMPLE-BUCKET). Alle IPv6-Adressen liegen außerhalb des zulässigen Bereichs, deshalb verhindert diese Richtlinie, dass IPv6-Adressen auf zugreife DOC-EXAMPLE-BUCKET.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "IPAllow",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3outposts:*",
    "Resource": "arn:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
    }
  }
]
}

```

Sie können das `-ConditionElement` der S3-on-Outposts-Bucket-Richtlinie ändern, um sowohl IPv4- (54.240.143.0/24) als auch IPv6-Adressbereiche (2001:DB8:1234:5678::/64) zuzulassen, wie im folgenden Beispiel gezeigt. Sie können denselben Typ `Condition-Block` verwenden, wie im Beispiel gezeigt, um Ihre IAM-Benutzer- und Bucket-Richtlinien zu aktualisieren.

```

"Condition": {
  "IpAddress": {
    "aws:SourceIp": [
      "54.240.143.0/24",
      "2001:DB8:1234:5678::/64"
    ]
  }
}
}

```

Bevor Sie IPv6 verwenden, müssen Sie alle relevanten IAM-Benutzer- und S3-Bucket-Richtlinien aktualisieren, die eine IP-Adressfilterung verwenden, um die IPv6-Adressbereiche zu berücksichtigen. Wir empfehlen Ihnen, Ihre IAM-Richtlinien mit den IPv6-Adressbereichen Ihres Unternehmens zu aktualisieren, ebenso wie mit Ihren vorhandenen IPv4-Adressbereichen. Ein Beispiel für eine Bucket-Richtlinie, die den Zugriff über IPv6 und IPv4 gestattet, finden Sie unter [Beschränken des Zugriffs auf bestimmte IP-Adressen](#).

Sie können Ihre IAM-Benutzerrichtlinien mit der IAM-Konsole unter <https://console.aws.amazon.com/iam/> überprüfen. Weitere Informationen zu IAM finden Sie im [IAM-Benutzerhandbuch](#). Informationen zum Bearbeiten von S3-on-Outposts-Bucket-Richtlinien finden Sie unter [Hinzufügen oder Bearbeiten einer Bucket-Richtlinie für einen Amazon-S3-on-Outposts-Bucket](#).

Testen der IP-Adresskompatibilität

Wenn Sie eine Linux- oder Unix-Instance oder macOS X-Plattform verwenden, können Sie Ihren Zugriff auf einen Dual-Stack-Endpunkt über IPv6 testen. Um beispielsweise die Verbindung zu Amazon S3-on-Outposts-Endpunkten über IPv6 zu testen, verwenden Sie den `dig` Befehl :

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

Wenn Ihr Dual-Stack-Endpunkt über ein IPv6-Netzwerk ordnungsgemäß eingerichtet ist, gibt der `dig` Befehl die verbundenen IPv6-Adressen zurück. Beispielsweise:

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

```
2600:1f14:2588:4800:b3a9:1460:159f:ebce
```

```
2600:1f14:2588:4802:6df6:c1fd:ef8a:fc76
```

```
2600:1f14:2588:4801:d802:8ccf:4e04:817
```

Verwenden von IPv6 mit AWS PrivateLink

S3 on Outposts unterstützt das IPv6-Protokoll für -AWS PrivateLinkServices und -Endpunkte. Mit AWS PrivateLink Unterstützung für das IPv6-Protokoll können Sie über IPv6-Netzwerke eine Verbindung zu Service-Endpunkten innerhalb Ihrer VPC herstellen, entweder von lokalen oder anderen privaten Verbindungen. Die IPv6-Unterstützung für [AWS PrivateLink für S3 on Outposts](#) ermöglicht Ihnen auch die Integration AWS PrivateLink mit Dual-Stack-Endpunkten. Schritte zur Aktivierung von IPv6 für AWS PrivateLink finden Sie unter [Beschleunigen Sie Ihre IPv6-Einführung mit -AWS PrivateLinkServices und -Endpunkten](#).

Note

Informationen zum Aktualisieren des unterstützten IP-Adresstyps von IPv4 auf IPv6 finden Sie unter [Ändern des unterstützten IP-Adresstyps](#) im AWS PrivateLink -Benutzerhandbuch.

Verwenden von IPv6 mit AWS PrivateLink

Wenn Sie AWS PrivateLink mit IPv6 verwenden, müssen Sie einen IPv6- oder Dual-Stack-VPC-Schnittstellenendpunkt erstellen. Allgemeine Schritte zum Erstellen eines VPC-Endpunkts mithilfe der

AWS Management Console finden Sie unter [Zugriff auf einen -AWS-Service über einen Schnittstellen-VPC-Endpunkt](#) im AWS PrivateLink -Benutzerhandbuch.

AWS Management Console

Gehen Sie wie folgt vor, um einen Schnittstellen-VPC-Endpunkt zu erstellen, der eine Verbindung zu S3 on Outposts herstellt.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie Endpunkt erstellen.
4. Wählen Sie bei Service category (Servicekategorie) die Option AWS services (-Services) aus.
5. Wählen Sie für Servicename den Service S3 on Outposts aus (com.amazonaws.us-east-1.s3-outposts).
6. Wählen Sie für VPC die VPC aus, von der aus Sie auf S3 on Outposts zugreifen.
7. Wählen Sie für Subnetze ein Subnetz pro Availability Zone aus, von dem aus Sie auf S3 on Outposts zugreifen. Sie können nicht mehrere Subnetze aus derselben Availability Zone auswählen. Für jedes von Ihnen ausgewählte Subnetz wird eine neue Endpunktnetzwerkschnittstelle erstellt. Standardmäßig werden IP-Adressen aus den Subnetz-IP-Adressbereichen den Endpunktnetzwerkschnittstellen zugewiesen. Um eine IP-Adresse für eine Endpunkt-Netzwerkschnittstelle festzulegen, wählen Sie IP-Adressen festlegen und geben Sie eine IPv6-Adresse aus dem Subnetz-Adressbereich ein.
8. Wählen Sie als IP-Adresstyp Dualstack aus. Weisen Sie Ihren Endpunktnetzwerkschnittstellen sowohl IPv4- als auch IPv6-Adressen zu. Diese Option wird nur unterstützt, wenn alle ausgewählten Subnetze sowohl IPv4- als auch IPv6-Adressbereiche aufweisen.
9. Wählen Sie für Sicherheitsgruppen die Sicherheitsgruppen aus, die den Endpunktnetzwerkschnittstellen für den VPC-Endpunkt zugeordnet werden sollen. Standardmäßig ist die Standardsicherheitsgruppe der VPC zugeordnet.
10. Wählen Sie für Richtlinie Vollzugriff, um alle Operationen aller Prinzipale auf allen Ressourcen über den VPC-Endpunkt zuzulassen. Wählen Sie andernfalls Benutzerdefiniert, um eine VPC-Endpunktrichtlinie anzufügen, die die Berechtigungen steuert, die Prinzipale haben, um Aktionen für Ressourcen über den VPC-Endpunkt durchzuführen. Diese Option ist nur verfügbar, wenn der Service VPC-Endpunktrichtlinien unterstützt. Weitere Informationen finden Sie unter [Endpunktrichtlinien](#).

11. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
12. Wählen Sie Endpunkt erstellen.

Example – Bucket-Richtlinie für S3 on Outposts

Damit S3 on Outposts mit Ihren VPC-Endpunkten interagieren kann, können Sie dann Ihre S3-on-Outposts-Richtlinie wie folgt aktualisieren:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3-outposts:*",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

AWS CLI

Note

Um das IPv6-Netzwerk auf Ihrem VPC-Endpunkt zu aktivieren, müssen Sie für den SupportedIpAddressType Filter für S3 on Outposts IPv6 festgelegt haben.

Im folgenden Beispiel wird der `create-vpc-endpoint` Befehl verwendet, um einen neuen Dual-Stack-Schnittstellenendpunkt zu erstellen.

```
aws ec2 create-vpc-endpoint \
--vpc-id vpc-12345678 \
--vpc-endpoint-type Interface \
--service-name com.amazonaws.us-east-1.s3-outposts \
--subnet-id subnet-12345678 \
--security-group-id sg-12345678 \
--ip-address-type dualstack \
--dns-options "DnsRecordIpType=dualstack"
```

Abhängig von der AWS PrivateLink Servicekonfiguration müssen neu erstellte Endpunktverbindungen möglicherweise vom VPC-Endpunkt-Serviceanbieter akzeptiert werden, bevor sie verwendet werden können. Weitere Informationen finden Sie unter [Annehmen und Ablehnen von Endpunktverbindungsanforderungen](#) im AWS PrivateLink -Benutzerhandbuch.

Im folgenden Beispiel wird der `modify-vpc-endpoint` Befehl verwendet, um den reinen IPv-VPC-Endpunkt auf einen Dual-Stack-Endpunkt zu aktualisieren. Der Dual-Stack-Endpunkt ermöglicht den Zugriff sowohl auf das IPv4- als auch auf das IPv6-Netzwerk.

```
aws ec2 modify-vpc-endpoint \  
--vpc-endpoint-id vpce-12345678 \  
--add-subnet-ids subnet-12345678 \  
--remove-subnet-ids subnet-12345678 \  
--ip-address-type dualstack \  
--dns-options "DnsRecordIpType=dualstack"
```

Weitere Informationen zum Aktivieren des IPv6-Netzwerks für finden Sie AWS PrivateLinkunter [Beschleunigen Sie Ihre IPv6-Einführung mit -AWS PrivateLinkServices und -Endpunkten](#).

Verwenden von S3-on-Outposts-Dual-Stack-Endpunkten

S3-on-Outposts-Dual-Stack-Endpunkte unterstützen Anfragen an S3-on-Outposts-Buckets über IPv6 und IPv4. In diesem Abschnitt wird beschrieben, wie Sie S3-on-Outposts-Dual-Stack-Endpunkte verwenden.

Themen

- [S3-on-Outposts-Dual-Stack-Endpunkte](#)
- [Verwenden von Dual-Stack-Endpunkten von der AWS CLI](#)
- [Verwenden von S3-on-Outposts-Dual-Stack-Endpunkten über die AWS -SDKs](#)

S3-on-Outposts-Dual-Stack-Endpunkte

Wenn Sie eine Anfrage an einen Dual-Stack-Endpunkt stellen, wird die S3-on-Outposts-Bucket-URL in eine IPv6- oder eine IPv4-Adresse aufgelöst. Weitere Informationen zum Zugriff auf einen S3-on-Outposts-Bucket über IPv6 finden Sie unter [Senden von Anforderungen an S3 on Outposts über IPv6](#).

Um über einen Dual-Stack-Endpoint auf einen S3-on-Outposts-Bucket zuzugreifen, verwenden Sie einen Endpunktnamen im Pfadformat. S3 on Outposts unterstützt nur regionale Dual-Stack-Endpunktnamen, was bedeutet, dass Sie die Region als Teil des Namens angeben müssen.

Verwenden Sie für einen Dual-Stack-FIPS-Endpoint im Pfadformat die folgende Namenskonvention:

```
s3-outposts-fips.region.api.aws
```

Verwenden Sie für Dual-Stack-Nicht-FIPS-Endpunkte die folgende Namenskonvention:

```
s3-outposts.region.api.aws
```

Note

Namen von Endpunkten im virtuellen Hosting-Stil werden in S3 on Outposts nicht unterstützt.

Verwenden von Dual-Stack-Endpunkten von der AWS CLI

Dieser Abschnitt enthält Beispiele für AWS CLI-Befehle für Anfragen an einen Dual-Stack-Endpoint. Weitere Informationen zum Einrichten der AWS CLI finden Sie unter [Erste Schritte mit dem AWS CLI und SDK for Java](#).

Sie legen den Konfigurationswert `true` in einem Profil in Ihrer `-AWS ConfigDatei` `use_dualstack_endpoint` auf fest, um alle Amazon S3-Anforderungen von den `s3api` AWS CLI Befehlen `s3` und an den Dual-Stack-Endpoint für die angegebene Region weiterzuleiten. Sie geben die Region in der Konfigurationsdatei oder in einem Befehl mit der `--region` Option an.

Bei der Verwendung von Dual-Stack-Endpunkten mit der `path` Adressierungsstil unterstützt. Der in der Konfigurationsdatei festgelegte Adressierungsstil bestimmt, ob der Bucket-Name im Hostnamen oder in der URL enthalten ist. Weitere Informationen finden Sie unter [s3outposts](#) im AWS CLI-Benutzerhandbuch.

Um einen Dual-Stack-Endpoint über die zu verwenden AWS CLI, verwenden Sie den `---endpoint-url` Parameter mit dem `- http://s3.dualstack.region.amazonaws.com` oder `-https://s3-outposts-fips.region.api.aws` Endpoint für alle `s3control-` oder `-s3outposts` Befehle.

Beispielsweise:

```
$ aws s3control list-regional-buckets --endpoint-url https://s3-outposts.region.api.aws
```

Verwenden von S3-on-Outposts-Dual-Stack-Endpunkten über die AWS -SDKs

Dieser Abschnitt enthält Beispiele für den Zugriff auf einen Dual-Stack-Endpunkt unter Verwendung der AWS-SDKs.

AWS SDK for Java 2.x Beispiel für einen -Dual-Stack-Endpunkt

Die folgenden Beispiele zeigen, wie Sie die `S3OutpostsClient` Klassen `S3ControlClient` und verwenden, um Dual-Stack-Endpunkte zu aktivieren, wenn Sie einen S3-on-Outposts-Client mit der erstellen AWS SDK for Java 2.x. Anweisungen zum Erstellen und Testen eines funktionierenden Java-Beispiels für Amazon S3 on Outposts finden Sie unter [Erste Schritte mit dem AWS CLI und SDK for Java](#).

Example – Erstellen einer `-S3ControlClient` Klasse mit aktivierten Dual-Stack-Endpunkten

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsRequest;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsResponse;
import software.amazon.awssdk.services.s3control.model.S3ControlException;

public class DualStackEndpointsExample1 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");
        String accountId = "111122223333";
        String navyId = "9876543210";

        try {
            // Create an S3ControlClient with dual-stack endpoints enabled.
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                .region(clientRegion)
                .dualstackEnabled(true)
                .build();

            ListRegionalBucketsRequest listRegionalBucketsRequest =
                ListRegionalBucketsRequest.builder()
```

```

        .accountId(accountId)

        .outpostId(navyId)

        .build();

        ListRegionalBucketsResponse listBuckets =
s3ControlClient.listRegionalBuckets(listRegionalBucketsRequest);
        System.out.printf("ListRegionalBuckets Response: %s%n",
listBuckets.toString());
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 on Outposts
couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch (S3ControlException e) {
        // Unknown exceptions will be thrown as an instance of this type.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 on Outposts couldn't be contacted for a response, or the
client
        // couldn't parse the response from Amazon S3 on Outposts.
        e.printStackTrace();
    }
}
}
}

```

Example – Erstellen eines **S3OutpostsClient** mit aktivierten Dual-Stack-Endpunkten

```

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3outposts.S3OutpostsClient;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsRequest;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsResponse;
import software.amazon.awssdk.services.s3outposts.model.S3OutpostsException;

public class DualStackEndpointsExample2 {

    public static void main(String[] args) {

```

```
Region clientRegion = Region.of("us-east-1");

try {
    // Create an S3OutpostsClient with dual-stack endpoints enabled.
    S3OutpostsClient s3OutpostsClient = S3OutpostsClient.builder()
                                                    .region(clientRegion)
                                                    .dualstackEnabled(true)
                                                    .build();

    ListEndpointsRequest listEndpointsRequest =
ListEndpointsRequest.builder().build();

    ListEndpointsResponse listEndpoints =
s3OutpostsClient.listEndpoints(listEndpointsRequest);
    System.out.printf("ListEndpoints Response: %s%n",
listEndpoints.toString());
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 on Outposts
couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
}
catch (S3OutpostsException e) {
    // Unknown exceptions will be thrown as an instance of this type.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 on Outposts couldn't be contacted for a response, or the
client
    // couldn't parse the response from Amazon S3 on Outposts.
    e.printStackTrace();
}
}
```

Wenn Sie die AWS SDK for Java 2.x unter Windows verwenden, müssen Sie möglicherweise die folgende Java Virtual Machine (JVM)-Eigenschaft festlegen:

```
java.net.preferIPv6Addresses=true
```

Codebeispiele für Amazon S3 mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie Amazon S3 mit einem verwenden AWS Software-Entwicklungskit (SDK).

Basics sind Codebeispiele, die Ihnen zeigen, wie Sie die wesentlichen Operationen innerhalb eines Dienstes ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben erledigen können, indem Sie mehrere Funktionen innerhalb eines Dienstes oder in Kombination mit anderen aufrufen AWS-Services.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte

Hello Amazon S3

Die folgenden Codebeispiele veranschaulichen die ersten Schritte mit Amazon S3.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

Code für die CMakeLists CMake .txt-Datei.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)
```

```
# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS s3)

# Set this project's name.
project("hello_s3")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
  libraries for the AWS SDK.
  string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
    "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
  list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
  # Copy relevant AWS SDK for C++ libraries into the current binary directory
  for running and debugging.

  # set(BIN_SUB_DIR "/Debug") # if you are building from the command line you
  may need to uncomment this
  # and set the proper subdirectory to the executables' location.

  AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
    ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
  hello_s3.cpp)

target_link_libraries(${PROJECT_NAME}
  ${AWSSDK_LINK_LIBRARIES})
```

Code für die Quelldatei „hello_s3.cpp“.

```
#include <aws/core/Aws.h>
#include <aws/s3/S3Client.h>
#include <iostream>
#include <aws/core/auth/AWSCredentialsProviderChain.h>
using namespace Aws;
using namespace Aws::Auth;

/*
 * A "Hello S3" starter application which initializes an Amazon Simple Storage
 * Service (Amazon S3) client
 * and lists the Amazon S3 buckets in the selected region.
 *
 * main function
 *
 * Usage: 'hello_s3'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
    // options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        // You don't normally have to test that you are authenticated. But the
        // S3 service permits anonymous requests, thus the s3Client will return "success"
        // and 0 buckets even if you are unauthenticated, which can be confusing to a new
        // user.

        auto provider =
        Aws::MakeShared<DefaultAWSCredentialsProviderChain>("alloc-tag");
        auto creds = provider->GetAWSCredentials();
        if (creds.IsEmpty()) {
            std::cerr << "Failed authentication" << std::endl;
        }

        Aws::S3::S3Client s3Client(clientConfig);
        auto outcome = s3Client.ListBuckets();
    }
}
```

```
    if (!outcome.IsSuccess()) {
        std::cerr << "Failed with error: " << outcome.GetError() <<
std::endl;
        result = 1;
    } else {
        std::cout << "Found " << outcome.GetResult().GetBuckets().size()
            << " buckets\n";
        for (auto &bucket: outcome.GetResult().GetBuckets()) {
            std::cout << bucket.GetName() << std::endl;
        }
    }
}

Aws::ShutdownAPI(options); // Should only be called once.
return result;
}
```

- API Einzelheiten finden Sie unter [ListBuckets](#) AWS SDK for C++ API Referenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
package main

import (
    "context"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Storage Service
```



```
// (Amazon S3) client and list up to 10 buckets in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS
account?")
        fmt.Println(err)
        return
    }
    s3Client := s3.NewFromConfig(sdkConfig)
    count := 10
    fmt.Printf("Let's list up to %v buckets for your account.\n", count)
    result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
    if err != nil {
        fmt.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
        return
    }
    if len(result.Buckets) == 0 {
        fmt.Println("You don't have any buckets!")
    } else {
        if count > len(result.Buckets) {
            count = len(result.Buckets)
        }
        for _, bucket := range result.Buckets[:count] {
            fmt.Printf("\t\t%v\n", *bucket.Name)
        }
    }
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu. GitHub Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class HelloS3 {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        listBuckets(s3);
    }

    public static void listBuckets(S3Client s3) {
        try {
            ListBucketsResponse response = s3.listBuckets();
            List<Bucket> bucketList = response.buckets();
            bucketList.forEach(bucket -> {
                System.out.println("Bucket Name: " + bucket.name());
            });
        }
    }
}
```

```
    });

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";

// When no region or credentials are provided, the SDK will use the
// region and credentials from the local AWS config.
const client = new S3Client({});

export const helloS3 = async () => {
    const command = new ListBucketsCommand({});

    const { Buckets } = await client.send(command);
    console.log("Buckets: ");
    console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
    return Buckets;
};
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for JavaScript API-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
use Aws\S3\S3Client;

$client = new S3Client(['region' => 'us-west-2']);
$results = $client->listBuckets();
var_dump($results);
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for PHP API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
import boto3

def hello_s3():
    """
    Use the AWS SDK for Python (Boto3) to create an Amazon Simple Storage Service
    (Amazon S3) resource and list the buckets in your account.
    This example uses the default settings specified in your shared credentials
    and config files.
    """
    s3_resource = boto3.resource("s3")
```

```
print("Hello, Amazon S3! Let's list your buckets:")
for bucket in s3_resource.buckets.all():
    print(f"\t{bucket.name}")

if __name__ == "__main__":
    hello_s3()
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
# frozen_string_literal: true

# S3Manager is a class responsible for managing S3 operations
# such as listing all S3 buckets in the current AWS account.
class S3Manager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all S3 buckets in the current AWS account.
  def list_buckets
    @logger.info('Here are the buckets in your account:')

    response = @client.list_buckets

    if response.buckets.empty?
      @logger.info("You don't have any S3 buckets yet.")
    else
      response.buckets.each do |bucket|
```

```
        @logger.info("- #{bucket.name}")
      end
    end
  rescue Aws::Errors::ServiceError => e
    @logger.error("Encountered an error while listing buckets: #{e.message}")
  end
end

if $PROGRAM_NAME == __FILE__
  s3_client = Aws::S3::Client.new
  manager = S3Manager.new(s3_client)
  manager.list_buckets
end
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
/// S3 Hello World Example using the AWS SDK for Rust.
///
/// This example lists the objects in a bucket, uploads an object to that bucket,
/// and then retrieves the object and prints some S3 information about the
/// object.
/// This shows a number of S3 features, including how to use built-in paginators
/// for large data sets.
///
/// # Arguments
///
/// * `client` - an S3 client configured appropriately for the environment.
/// * `bucket` - the bucket name that the object will be uploaded to. Must be
/// present in the region the `client` is configured to use.
/// * `filename` - a reference to a path that will be read and uploaded to S3.
```

```
/// * `key` - the string key that the object will be uploaded as inside the
bucket.
async fn list_bucket_and_upload_object(
    client: &aws_sdk_s3::Client,
    bucket: &str,
    filepath: &Path,
    key: &str,
) -> Result<(), S3ExampleError> {
    // List the buckets in this account
    let mut objects = client
        .list_objects_v2()
        .bucket(bucket)
        .into_paginator()
        .send();

    println!("key\tetag\tlast_modified\tstorage_class");
    while let Some(Ok(object)) = objects.next().await {
        for item in object.contents() {
            println!(
                "{}\t{}\t{}\t{}",
                item.key().unwrap_or_default(),
                item.e_tag().unwrap_or_default(),
                item.last_modified()
                    .map(|lm| format!("{lm}"))
                    .unwrap_or_default(),
                item.storage_class()
                    .map(|sc| format!("{sc}"))
                    .unwrap_or_default()
            );
        }
    }

    // Prepare a ByteStream around the file, and upload the object using that
    ByteStream.
    let body = aws_sdk_s3::primitives::ByteStream::from_path(filepath)
        .await
        .map_err(|err| {
            S3ExampleError::new(format!(
                "Failed to create bytestream for {filepath:?} ({err:?})"
            ))
        })?;
    let resp = client
        .put_object()
        .bucket(bucket)
```

```

        .key(key)
        .body(body)
        .send()
        .await?;

println!(
    "Upload success. Version: {:?}" ,
    resp.version_id()
        .expect("S3 Object upload missing version ID")
);

// Retrieve the just-uploaded object.
let resp = client.get_object().bucket(bucket).key(key).send().await?;
println!("etag: {}", resp.e_tag().unwrap_or("(missing)"));
println!("version: {}", resp.version_id().unwrap_or("(missing)"));

Ok(())
}

```

ExampleError S3-Dienstprogramme

```

/// S3ExampleError provides a From<T: ProvideErrorMetadata> impl to extract
/// client-specific error details. This serves as a consistent backup to handling
/// specific service errors, depending on what is needed by the scenario.
/// It is used throughout the code examples for the AWS SDK for Rust.
#[derive(Debug)]
pub struct S3ExampleError(String);
impl S3ExampleError {
    pub fn new(value: impl Into<String>) -> Self {
        S3ExampleError(value.into())
    }

    pub fn add_message(self, message: impl Into<String>) -> Self {
        S3ExampleError(format!("{}: {}", message.into(), self.0))
    }
}

impl<T: aws_sdk_s3::error::ProvideErrorMetadata> From<T> for S3ExampleError {
    fn from(value: T) -> Self {
        S3ExampleError(format!(
            "{}: {}",
            value

```



```
        .code()
        .map(String::from)
        .unwrap_or("unknown code".into()),
    value
        .message()
        .map(String::from)
        .unwrap_or("missing reason".into()),
    ))
}
}

impl std::error::Error for S3ExampleError {}

impl std::fmt::Display for S3ExampleError {
    fn fmt(&self, f: &mut std::fmt::Formatter<'_>) -> std::fmt::Result {
        write!(f, "{}", self.0)
    }
}
```

- API Einzelheiten finden Sie [ListBuckets](#) in AWS SDK als API Referenz auf Rust.

Codebeispiele

- [Grundlegende Beispiele für die Verwendung von Amazon S3 AWS SDKs](#)
 - [Hello Amazon S3](#)
 - [Lernen Sie die Grundlagen von Amazon S3 mit einem AWS SDK](#)
 - [Aktionen für Amazon S3 mit AWS SDKs](#)
 - [Verwenden Sie AbortMultipartUpload mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie AbortMultipartUploads mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CompleteMultipartUpload mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CopyObject mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CreateBucket mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CreateMultiRegionAccessPoint mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CreateMultipartUpload mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucket mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketAnalyticsConfiguration mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketCors mit einem AWS SDKoder CLI](#)

- [Verwenden Sie DeleteBucketEncryption mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketInventoryConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketLifecycle mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketMetricsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketWebsite mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteObjectTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteObjects mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeletePublicAccessBlock mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAccelerateConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAnalyticsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketCors mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketEncryption mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketInventoryConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLifecycleConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLocation mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLogging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketMetricsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketNotification mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketPolicyStatus mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketRequestPayment mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketVersioning mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketWebsite mit einem AWS SDKoder CLI](#)

- [Verwenden Sie GetObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectAttributes mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectLegalHold mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectLockConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectRetention mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetPublicAccessBlock mit einem AWS SDKoder CLI](#)
- [Verwenden Sie HeadBucket mit einem AWS SDKoder CLI](#)
- [Verwenden Sie HeadObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBucketAnalyticsConfigurations mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBucketInventoryConfigurations mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBuckets mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListMultipartUploads mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjectVersions mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjects mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjectsV2 mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketAccelerateConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketCors mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketEncryption mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketLifecycleConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketLogging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketNotification mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketNotificationConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketRequestPayment mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketVersioning mit einem AWS SDKoder CLI](#)

- [Verwenden Sie PutBucketWebsite mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectLegalHold mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectLockConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectRetention mit einem AWS SDKoder CLI](#)
- [Verwenden Sie RestoreObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie SelectObjectContent mit einem AWS SDKoder CLI](#)
- [Verwenden Sie UploadPart mit einem AWS SDKoder CLI](#)
- [Szenarien für die Verwendung von Amazon S3 AWS SDKs](#)
 - [Konvertieren Sie Text in Sprache und zurück in Text mit einem AWS SDK](#)
 - [Erstellen Sie eine URL für Amazon S3 vorsignierte Datei mit einem AWS SDK](#)
 - [Eine Anwendung für Foto-Asset-Management erstellen, mit der Benutzer Fotos mithilfe von Labels verwalten können](#)
 - [Eine Webseite, die Amazon S3 S3-Objekte mithilfe eines auflistet AWS SDK](#)
 - [Erstellen Sie eine Amazon-Textextract-Explorer-Anwendung](#)
 - [Löschen Sie alle Objekte in einem bestimmten Amazon S3 S3-Bucket mit einem AWS SDK.](#)
 - [Löschen Sie unvollständige mehrteilige Uploads auf Amazon S3 mit einem AWS SDK](#)
 - [Erkennung PPE in Bildern mit Amazon Rekognition mithilfe eines AWS SDK](#)
 - [Erkennen Sie Entitäten in Text, der aus einem Bild extrahiert wurde, mithilfe eines AWS SDK](#)
 - [Erkennen Sie Gesichter in einem Bild mit einem AWS SDK](#)
 - [Erkennen Sie Objekte in Bildern mit Amazon Rekognition mithilfe eines AWS SDK](#)
 - [Erkennen Sie Personen und Objekte in einem Video mit Amazon Rekognition mithilfe eines AWS SDK](#)
 - [Herunterladen aller Objekte aus einem Amazon Simple Storage Service \(Amazon S3\)-Bucket in ein lokales Verzeichnis](#)
 - [Rufen Sie ein Amazon S3 S3-Objekt von einem Access Point mit mehreren Regionen ab, indem Sie eine AWS SDK](#)
 - [Holen Sie sich ein Objekt aus einem Amazon S3 S3-Bucket mit einem AWS SDK, unter Angabe eines If-Modified-Since-Headers](#)
- [Erste Schritte mit der Verschlüsselung von Amazon S3 S3-Objekten mithilfe eines AWS SDK](#)

- [Erste Schritte mit Tags für Amazon S3 S3-Objekte mithilfe eines AWS SDK](#)
- [Arbeiten Sie mit den Objektsperrefunktionen von Amazon S3 mithilfe eines AWS SDK](#)
- [Verwalten Sie Zugriffskontrolllisten \(ACLs\) für Amazon S3 S3-Buckets mit einem AWS SDK](#)
- [Verwalten Sie versionierte Amazon S3 S3-Objekte stapelweise mit einer Lambda-Funktion mithilfe eines AWS SDK](#)
- [Analysieren Sie Amazon S3 URIs mit einem AWS SDK](#)
- [Erstellen Sie eine mehrteilige Kopie eines Amazon S3 S3-Objekts mit einem AWS SDK](#)
- [Führen Sie einen mehrteiligen Upload eines Amazon S3 S3-Objekts mit einem AWS SDK](#)
- [Empfangen und verarbeiten Sie Amazon S3 S3-Ereignisbenachrichtigungen mithilfe eines AWS SDK.](#)
- [Speichern EXIF und andere Bildinformationen mithilfe eines AWS SDK](#)
- [Senden Sie S3-Ereignisbenachrichtigungen an Amazon EventBridge mit einem AWS SDK](#)
- [Verfolgen Sie den Upload oder Download eines Amazon S3 S3-Objekts mithilfe eines AWS SDK](#)
- [Transformieren Sie Daten für Ihre Anwendung mit S3 Object Lambda](#)
- [Beispielansätze für Einheiten- und Integrationstests mit einem AWS SDK](#)
- [Rekursives Hochladen eines lokalen Verzeichnisses in einen Amazon Simple Storage Service \(Amazon S3\)-Bucket](#)
- [Laden Sie große Dateien zu und von Amazon S3 hoch oder laden Sie sie herunter mit einem AWS SDK](#)
- [Laden Sie einen Stream unbekannter Größe mit einem in ein Amazon S3 S3-Objekt hoch AWS SDK](#)
- [Verwenden Sie Prüfsummen, um mit einem Amazon S3 S3-Objekt zu arbeiten, indem Sie AWS SDK](#)
- [Arbeiten Sie mit den Objektintegritätsfunktionen von Amazon S3 mithilfe eines AWS SDK](#)
- [Arbeiten Sie mit Objekten mit Amazon S3 S3-Version unter Verwendung eines AWS SDK](#)
- [Serverlose Beispiele für Amazon S3 mit AWS SDKs](#)
 - [Aufrufen einer Lambda-Funktion über einen Amazon-S3-Auslöser](#)

Grundlegende Beispiele für die Verwendung von Amazon S3 AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie die Grundlagen von Amazon Simple Storage Service verwenden können mit AWS SDKs.

Beispiele

- [Hello Amazon S3](#)
- [Lernen Sie die Grundlagen von Amazon S3 mit einem AWS SDK](#)
- [Aktionen für Amazon S3 mit AWS SDKs](#)
 - [Verwenden Sie AbortMultipartUpload mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie AbortMultipartUploads mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CompleteMultipartUpload mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CopyObject mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CreateBucket mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CreateMultiRegionAccessPoint mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie CreateMultipartUpload mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucket mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketAnalyticsConfiguration mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketCors mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketEncryption mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketInventoryConfiguration mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketLifecycle mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketMetricsConfiguration mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketPolicy mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketReplication mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketTagging mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteBucketWebsite mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteObject mit einem AWS SDKoder CLI](#)
 - [Verwenden Sie DeleteObjectTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteObjects mit einem AWS SDKoder CLI](#)

- [Verwenden Sie DeletePublicAccessBlock mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAccelerateConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAnalyticsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketCors mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketEncryption mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketInventoryConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLifecycleConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLocation mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLogging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketMetricsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketNotification mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketPolicyStatus mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketRequestPayment mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketVersioning mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketWebsite mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectAttributes mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectLegalHold mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectLockConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectRetention mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetPublicAccessBlock mit einem AWS SDKoder CLI](#)
- [Verwenden Sie HeadBucket mit einem AWS SDKoder CLI](#)
- [Verwenden Sie HeadObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBucketAnalyticsConfigurations mit einem AWS SDKoder CLI](#)

- [Verwenden Sie ListBucketInventoryConfigurations mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBuckets mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListMultipartUploads mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjectVersions mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjects mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjectsV2 mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketAccelerateConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketCors mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketEncryption mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketLifecycleConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketLogging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketNotification mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketNotificationConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketRequestPayment mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketVersioning mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketWebsite mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectLegalHold mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectLockConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectRetention mit einem AWS SDKoder CLI](#)
- [Verwenden Sie RestoreObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie SelectObjectContent mit einem AWS SDKoder CLI](#)
- [Verwenden Sie UploadPart mit einem AWS SDKoder CLI](#)

Hello Amazon S3

Die folgenden Codebeispiele veranschaulichen die ersten Schritte mit Amazon S3.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

Code für die CMakeLists CMake .txt-Datei.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS s3)

# Set this project's name.
project("hello_s3")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
  libraries for the AWS SDK.
  string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
    "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
  list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})
```

```
if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
    # Copy relevant AWS SDK for C++ libraries into the current binary directory
    for running and debugging.

    # set(BIN_SUB_DIR "/Debug") # if you are building from the command line you
    may need to uncomment this
    # and set the proper subdirectory to the executables' location.

    AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
    ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
    hello_s3.cpp)

target_link_libraries(${PROJECT_NAME}
    ${AWSSDK_LINK_LIBRARIES})
```

Code für die Quelldatei „hello_s3.cpp“.

```
#include <aws/core/Aws.h>
#include <aws/s3/S3Client.h>
#include <iostream>
#include <aws/core/auth/AWSCredentialsProviderChain.h>
using namespace Aws;
using namespace Aws::Auth;

/*
 * A "Hello S3" starter application which initializes an Amazon Simple Storage
 * Service (Amazon S3) client
 * and lists the Amazon S3 buckets in the selected region.
 *
 * main function
 *
 * Usage: 'hello_s3'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
    // options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
```

```
Aws::InitAPI(options); // Should only be called once.
int result = 0;
{
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

    // You don't normally have to test that you are authenticated. But the
    // S3 service permits anonymous requests, thus the s3Client will return "success"
    // and 0 buckets even if you are unauthenticated, which can be confusing to a new
    // user.

    auto provider =
    Aws::MakeShared<DefaultAWSCredentialsProviderChain>("alloc-tag");
    auto creds = provider->GetAWSCredentials();
    if (creds.IsEmpty()) {
        std::cerr << "Failed authentication" << std::endl;
    }

    Aws::S3::S3Client s3Client(clientConfig);
    auto outcome = s3Client.ListBuckets();

    if (!outcome.IsSuccess()) {
        std::cerr << "Failed with error: " << outcome.GetError() <<
std::endl;
        result = 1;
    } else {
        std::cout << "Found " << outcome.GetResult().GetBuckets().size()
        << " buckets\n";
        for (auto &bucket: outcome.GetResult().GetBuckets()) {
            std::cout << bucket.GetName() << std::endl;
        }
    }
}

Aws::ShutdownAPI(options); // Should only be called once.
return result;
}
```

- API Einheiten finden Sie unter [ListBuckets](#) AWS SDK for C++ API Referenz.

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
package main

import (
    "context"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Storage Service
// (Amazon S3) client and list up to 10 buckets in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    sdkConfig, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS
account?")
        fmt.Println(err)
        return
    }
    s3Client := s3.NewFromConfig(sdkConfig)
    count := 10
    fmt.Printf("Let's list up to %v buckets for your account.\n", count)
    result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
    if err != nil {
        fmt.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
        return
    }
    if len(result.Buckets) == 0 {
```

```
    fmt.Println("You don't have any buckets!")
} else {
    if count > len(result.Buckets) {
        count = len(result.Buckets)
    }
    for _, bucket := range result.Buckets[:count] {
        fmt.Printf("\t\t%v\n", *bucket.Name)
    }
}
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie in [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class HelloS3 {
```

```
public static void main(String[] args) {
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    listBuckets(s3);
}

public static void listBuckets(S3Client s3) {
    try {
        ListBucketsResponse response = s3.listBuckets();
        List<Bucket> bucketList = response.buckets();
        bucketList.forEach(bucket -> {
            System.out.println("Bucket Name: " + bucket.name());
        });
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";

// When no region or credentials are provided, the SDK will use the
// region and credentials from the local AWS config.
const client = new S3Client({});
```

```
export const helloS3 = async () => {
  const command = new ListBucketsCommand({});

  const { Buckets } = await client.send(command);
  console.log("Buckets: ");
  console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
  return Buckets;
};
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for JavaScript API-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
use Aws\S3\S3Client;

$client = new S3Client(['region' => 'us-west-2']);
$results = $client->listBuckets();
var_dump($results);
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for PHP API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import boto3

def hello_s3():
    """
    Use the AWS SDK for Python (Boto3) to create an Amazon Simple Storage Service
    (Amazon S3) resource and list the buckets in your account.
    This example uses the default settings specified in your shared credentials
    and config files.
    """
    s3_resource = boto3.resource("s3")
    print("Hello, Amazon S3! Let's list your buckets:")
    for bucket in s3_resource.buckets.all():
        print(f"\t{bucket.name}")

if __name__ == "__main__":
    hello_s3()
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK für Python (Boto3) API -Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).


```
# frozen_string_literal: true

# S3Manager is a class responsible for managing S3 operations
# such as listing all S3 buckets in the current AWS account.
class S3Manager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all S3 buckets in the current AWS account.
  def list_buckets
    @logger.info('Here are the buckets in your account:')

    response = @client.list_buckets

    if response.buckets.empty?
      @logger.info("You don't have any S3 buckets yet.")
    else
      response.buckets.each do |bucket|
        @logger.info("- #{bucket.name}")
      end
    end
  end

  rescue Aws::Errors::ServiceError => e
    @logger.error("Encountered an error while listing buckets: #{e.message}")
  end
end

if $PROGRAM_NAME == __FILE__
  s3_client = Aws::S3::Client.new
  manager = S3Manager.new(s3_client)
  manager.list_buckets
end
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// S3 Hello World Example using the AWS SDK for Rust.
///
/// This example lists the objects in a bucket, uploads an object to that bucket,
/// and then retrieves the object and prints some S3 information about the
/// object.
/// This shows a number of S3 features, including how to use built-in paginators
/// for large data sets.
///
/// # Arguments
///
/// * `client` - an S3 client configured appropriately for the environment.
/// * `bucket` - the bucket name that the object will be uploaded to. Must be
///   present in the region the `client` is configured to use.
/// * `filename` - a reference to a path that will be read and uploaded to S3.
/// * `key` - the string key that the object will be uploaded as inside the
///   bucket.
async fn list_bucket_and_upload_object(
    client: &aws_sdk_s3::Client,
    bucket: &str,
    filepath: &Path,
    key: &str,
) -> Result<(), S3ExampleError> {
    // List the buckets in this account
    let mut objects = client
        .list_objects_v2()
        .bucket(bucket)
        .into_paginator()
        .send();

    println!("key\tetag\tlast_modified\tstorage_class");
    while let Some(Ok(object)) = objects.next().await {
        for item in object.contents() {
```

```
        println!(
            "{}\t{}\t{}\t{}",
            item.key().unwrap_or_default(),
            item.e_tag().unwrap_or_default(),
            item.last_modified()
                .map(|lm| format!("{lm}"))
                .unwrap_or_default(),
            item.storage_class()
                .map(|sc| format!("{sc}"))
                .unwrap_or_default()
        );
    }
}

// Prepare a ByteStream around the file, and upload the object using that
// ByteStream.
let body = aws_sdk_s3::primitives::ByteStream::from_path(filepath)
    .await
    .map_err(|err| {
        S3ExampleError::new(format!(
            "Failed to create bytestream for {filepath:?} ({err:?})"
        ))
    })?;
let resp = client
    .put_object()
    .bucket(bucket)
    .key(key)
    .body(body)
    .send()
    .await?;

println!(
    "Upload success. Version: {:?}",
    resp.version_id()
        .expect("S3 Object upload missing version ID")
);

// Retrieve the just-uploaded object.
let resp = client.get_object().bucket(bucket).key(key).send().await?;
println!("etag: {}", resp.e_tag().unwrap_or("missing"));
println!("version: {}", resp.version_id().unwrap_or("missing"));

Ok(())
}
```

ExampleError S3-Dienstprogramme

```
/// S3ExampleError provides a From<T: ProvideErrorMetadata> impl to extract
/// client-specific error details. This serves as a consistent backup to handling
/// specific service errors, depending on what is needed by the scenario.
/// It is used throughout the code examples for the AWS SDK for Rust.
#[derive(Debug)]
pub struct S3ExampleError(String);
impl S3ExampleError {
    pub fn new(value: impl Into<String>) -> Self {
        S3ExampleError(value.into())
    }

    pub fn add_message(self, message: impl Into<String>) -> Self {
        S3ExampleError(format!("{}", message.into(), self.0))
    }
}

impl<T: aws_sdk_s3::error::ProvideErrorMetadata> From<T> for S3ExampleError {
    fn from(value: T) -> Self {
        S3ExampleError(format!(
            "{}: {}",
            value
                .code()
                .map(String::from)
                .unwrap_or("unknown code".into()),
            value
                .message()
                .map(String::from)
                .unwrap_or("missing reason".into()),
        ))
    }
}

impl std::error::Error for S3ExampleError {}

impl std::fmt::Display for S3ExampleError {
    fn fmt(&self, f: &mut std::fmt::Formatter<'_>) -> std::fmt::Result {
        write!(f, "{}", self.0)
    }
}
```

- API-Einheiten finden Sie [ListBuckets](#) in AWS SDK als API Referenz auf Rust.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Lernen Sie die Grundlagen von Amazon S3 mit einem AWS SDK

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Erstellen Sie einen Bucket und laden Sie eine Datei in ihn hoch.
- Laden Sie ein Objekt aus einem Bucket herunter.
- Kopieren Sie ein Objekt in einen Unterordner eines Buckets.
- Listen Sie die Objekte in einem Bucket auf.
- Löschen Sie die Bucket-Objekte und den Bucket.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
public class S3_Basics
{
    public static async Task Main()
    {
        // Create an Amazon S3 client object. The constructor uses the
        // default user installed on the system. To work with Amazon S3
        // features in a different AWS Region, pass the AWS Region as a
        // parameter to the client constructor.
        IAmazonS3 client = new AmazonS3Client();
        string bucketName = string.Empty;
        string filePath = string.Empty;
```

```
string keyName = string.Empty;

var sepBar = new string('-', Console.WindowWidth);

Console.WriteLine(sepBar);
Console.WriteLine("Amazon Simple Storage Service (Amazon S3) basic");
Console.WriteLine("procedures. This application will:");
Console.WriteLine("\n\t1. Create a bucket");
Console.WriteLine("\n\t2. Upload an object to the new bucket");
Console.WriteLine("\n\t3. Copy the uploaded object to a folder in the
bucket");
Console.WriteLine("\n\t4. List the items in the new bucket");
Console.WriteLine("\n\t5. Delete all the items in the bucket");
Console.WriteLine("\n\t6. Delete the bucket");
Console.WriteLine(sepBar);

// Create a bucket.
Console.WriteLine($"{sepBar}");
Console.WriteLine("\nCreate a new Amazon S3 bucket.\n");
Console.WriteLine(sepBar);

Console.Write("Please enter a name for the new bucket: ");
bucketName = Console.ReadLine();

var success = await S3Bucket.CreateBucketAsync(client, bucketName);
if (success)
{
    Console.WriteLine($"Successfully created bucket: {bucketName}.
\n");
}
else
{
    Console.WriteLine($"Could not create bucket: {bucketName}.\n");
}

Console.WriteLine(sepBar);
Console.WriteLine("Upload a file to the new bucket.");
Console.WriteLine(sepBar);

// Get the local path and filename for the file to upload.
while (string.IsNullOrEmpty(filePath))
{
    Console.Write("Please enter the path and filename of the file to
upload: ");
```

```
        filePath = Console.ReadLine();

        // Confirm that the file exists on the local computer.
        if (!File.Exists(filePath))
        {
            Console.WriteLine($"Couldn't find {filePath}. Try again.\n");
            filePath = string.Empty;
        }
    }

    // Get the file name from the full path.
    keyName = Path.GetFileName(filePath);

    success = await S3Bucket.UploadFileAsync(client, bucketName, keyName,
filePath);

    if (success)
    {
        Console.WriteLine($"Successfully uploaded {keyName} from
{filePath} to {bucketName}.\n");
    }
    else
    {
        Console.WriteLine($"Could not upload {keyName}.\n");
    }

    // Set the file path to an empty string to avoid overwriting the
    // file we just uploaded to the bucket.
    filePath = string.Empty;

    // Now get a new location where we can save the file.
    while (string.IsNullOrEmpty(filePath))
    {
        // First get the path to which the file will be downloaded.
        Console.Write("Please enter the path where the file will be
downloaded: ");
        filePath = Console.ReadLine();

        // Confirm that the file exists on the local computer.
        if (File.Exists($"{filePath}\\{keyName}"))
        {
            Console.WriteLine($"Sorry, the file already exists in that
location.\n");
            filePath = string.Empty;
        }
    }
}
```

```
    }
}

// Download an object from a bucket.
success = await S3Bucket.DownloadObjectFromBucketAsync(client,
bucketName, keyName, filePath);

if (success)
{
    Console.WriteLine($"Successfully downloaded {keyName}.\n");
}
else
{
    Console.WriteLine($"Sorry, could not download {keyName}.\n");
}

// Copy the object to a different folder in the bucket.
string folderName = string.Empty;

while (string.IsNullOrEmpty(folderName))
{
    Console.Write("Please enter the name of the folder to copy your
object to: ");
    folderName = Console.ReadLine();
}

while (string.IsNullOrEmpty(keyName))
{
    // Get the name to give to the object once uploaded.
    Console.Write("Enter the name of the object to copy: ");
    keyName = Console.ReadLine();
}

await S3Bucket.CopyObjectInBucketAsync(client, bucketName, keyName,
folderName);

// List the objects in the bucket.
await S3Bucket.ListBucketContentsAsync(client, bucketName);

// Delete the contents of the bucket.
await S3Bucket.DeleteBucketContentsAsync(client, bucketName);

// Deleting the bucket too quickly after deleting its contents will
// cause an error that the bucket isn't empty. So...
```



```
        Console.WriteLine("Press <Enter> when you are ready to delete the
bucket.");
        _ = Console.ReadLine();

        // Delete the bucket.
        await S3Bucket.DeleteBucketAsync(client, bucketName);
    }
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK for .NET API-Referenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#####
# function s3_getting_started
#
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#     0 - If successful.
```

```
#      1 - If an error occurred.
#####
function s3_getting_started() {
  {
    if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
      cd bucket-lifecycle-operations || exit

      source ./bucket_operations.sh
      cd ..
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the Amazon S3 getting started demo."
  echo_repeat "*" 88

  local bucket_name
  bucket_name=$(generate_random_name "amzn-s3-demo-bucket")

  local region_code
  region_code=$(aws configure get region)

  if create_bucket -b "$bucket_name" -r "$region_code"; then
    echo "Created demo bucket named $bucket_name"
  else
    errecho "The bucket failed to create. This demo will exit."
    return 1
  fi

  local file_name
  while [ -z "$file_name" ]; do
    echo -n "Enter a file you want to upload to your bucket: "
    get_input
    file_name=$get_input_result

    if [ ! -f "$file_name" ]; then
      echo "Could not find file $file_name. Are you sure it exists?"
      file_name=""
    fi
  done

  local key
  key="$(basename "$file_name")"
```

```
local result=0
if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
    echo "Uploaded file $file_name into bucket $bucket_name with key $key."
else
    result=1
fi

local destination_file
destination_file="$file_name.download"
if yes_no_input "Would you like to download $key to the file $destination_file?
(y/n) "; then
    if download_object_from_bucket "$bucket_name" "$destination_file" "$key";
then
        echo "Downloaded $key in the bucket $bucket_name to the file
$destination_file."
    else
        result=1
    fi
fi

if yes_no_input "Would you like to copy $key a new object key in your bucket?
(y/n) "; then
    local to_key
    to_key="demo/$key"
    if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
        echo "Copied $key in the bucket $bucket_name to the $to_key."
    else
        result=1
    fi
fi

local bucket_items
bucket_items=$(list_items_in_bucket "$bucket_name")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    result=1
fi

echo "Your bucket contains the following items."
echo -e "Name\t\tSize"
echo "$bucket_items"
```

```

if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
(y/n) "; then
    bucket_items=$(echo "$bucket_items" | cut -f 1)

    if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
        echo "The following items were deleted from the bucket $bucket_name"
        echo "$bucket_items"
    else
        result=1
    fi

    if delete_bucket "$bucket_name"; then
        echo "Deleted the bucket $bucket_name"
    else
        result=1
    fi
fi

return $result
}

```

Die in diesem Szenario verwendeten Amazon S3 S3-Funktionen.

```

#####
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#     -b bucket_name -- The name of the bucket to create.
#     -r region_code -- The code for an AWS Region in which to
#                       create the bucket.
#
# Returns:
#     The URL of the bucket that was created.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response

```

```
local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally
unique."
    echo "  [-r region_code]  The code for an AWS Region in which the bucket is
created."
    echo ""
}

# Retrieve the calling parameters.
while getopt "b:r:h" option; do
    case "${option}" in
        b) bucket_name="${OPTARG}" ;;
        r) region_code="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done

if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
fi

local bucket_config_arg
# A location constraint for "us-east-1" returns an error.
if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
```

```

iecho "    Bucket name:  $bucket_name"
iecho "    Region code:  $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
fi

# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
}

#####
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file to.
#     $2 - The path and file name of the local file to copy to the bucket.
#     $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_file_to_bucket() {
    local response bucket_name source_file destination_file_name
    bucket_name=$1
    source_file=$2
    destination_file_name=$3

    response=$(aws s3api put-object \
        --bucket "$bucket_name" \

```

```

--body "$source_file" \
--key "$destination_file_name")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
fi
}

#####
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#     $1 - The name of the bucket to download the object from.
#     $2 - The path and file name to store the downloaded bucket.
#     $3 - The key (name) of the object in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function download_object_from_bucket() {
    local bucket_name=$1
    local destination_file_name=$2
    local object_name=$3
    local response

    response=$(aws s3api get-object \
        --bucket "$bucket_name" \
        --key "$object_name" \
        "$destination_file_name")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
        errecho "ERROR: AWS reports put-object operation failed.\n$response"
        return 1
    fi
}

#####
# function copy_item_in_bucket

```

```

#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response

    response=$(aws s3api copy-object \
        --bucket "$bucket_name" \
        --copy-source "$bucket_name/$source_key" \
        --key "$destination_key")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
        return 1
    fi
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.
#
# And:
#     0 - If successful.

```



```

#      1 - If it fails.
#####
function list_items_in_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api list-objects \
        --bucket "$bucket_name" \
        --output text \
        --query 'Contents[].{Key: Key, Size: Size}')

    # shellcheck disable=SC2181
    if [[ ${?} -eq 0 ]]; then
        echo "$response"
    else
        errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
        return 1
    fi
}

#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - A list of keys in the bucket to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_items_in_bucket() {
    local bucket_name=$1
    local keys=$2
    local response

    # Create the JSON for the items to delete.
    local delete_items
    delete_items="{\"Objects\":["
    for key in $keys; do
        delete_items="$delete_items{\"Key\": \"$key\"},"
    done

```

```

delete_items=${delete_items%?} # Remove the final comma.
delete_items="$delete_items]}"

response=$(aws s3api delete-objects \
  --bucket "$bucket_name" \
  --delete "$delete_items")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
  errecho "ERROR: AWS reports s3api delete-object operation failed.\n
$response"
  return 1
fi
}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}

```

- API-Einheiten finden Sie in den folgenden Themen unter AWS CLI Befehlsreferenz.

- [CopyObject](#)
- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#include <iostream>
#include <aws/core/Aws.h>
#include <aws/s3/S3Client.h>
#include <aws/s3/model/CopyObjectRequest.h>
#include <aws/s3/model/CreateBucketRequest.h>
#include <aws/s3/model/DeleteBucketRequest.h>
#include <aws/s3/model/DeleteObjectRequest.h>
#include <aws/s3/model/GetObjectRequest.h>
#include <aws/s3/model/ListObjectsV2Request.h>
#include <aws/s3/model/PutObjectRequest.h>
#include <aws/s3/model/BucketLocationConstraint.h>
#include <aws/s3/model/CreateBucketConfiguration.h>
#include <aws/core/utils/UUID.h>
#include <aws/core/utils/StringUtils.h>
#include <aws/core/utils/memory/stl/AWSAllocator.h>
#include <fstream>
#include "s3_examples.h"

namespace AwsDoc {
    namespace S3 {
```

```
    //! Delete an S3 bucket.
    /*!
        \param bucketName: The S3 bucket's name.
        \param client: An S3 client.
        \return bool: Function succeeded.
    */
    static bool
    deleteBucket(const Aws::String &bucketName, Aws::S3::S3Client &client);

    //! Delete an object in an S3 bucket.
    /*!
        \param bucketName: The S3 bucket's name.
        \param key: The key for the object in the S3 bucket.
        \param client: An S3 client.
        \return bool: Function succeeded.
    */
    static bool
    deleteObjectFromBucket(const Aws::String &bucketName, const Aws::String
&key,
                            Aws::S3::S3Client &client);
    }
}

//! Scenario to create, copy, and delete S3 buckets and objects.
/*!
    \param uploadFilePath: Path to file to upload to an Amazon S3 bucket.
    \param saveFilePath: Path for saving a downloaded S3 object.
    \param clientConfig: Aws client configuration.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::S3_GettingStartedScenario(const Aws::String &uploadFilePath,
                                           const Aws::String &saveFilePath,
                                           const Aws::Client::ClientConfiguration
&clientConfig) {

    Aws::S3::S3Client client(clientConfig);

    // Create a unique bucket name which is only temporary and will be deleted.
    // Format: "amzn-s3-demo-bucket-" + lowercase UUID.
    Aws::String uuid = Aws::Utils::UUID::RandomUUID();
    Aws::String bucketName = "amzn-s3-demo-bucket-" +
        Aws::Utils::StringUtils::ToLower(uuid.c_str());
```

```
// 1. Create a bucket.
{
    Aws::S3::Model::CreateBucketRequest request;
    request.SetBucket(bucketName);

    if (clientConfig.region != Aws::Region::US_EAST_1) {
        Aws::S3::Model::CreateBucketConfiguration createBucketConfiguration;
        createBucketConfiguration.WithLocationConstraint(
            Aws::S3::Model::BucketLocationConstraintMapper::GetBucketLocationConstraintForName(
                clientConfig.region));
        request.WithCreateBucketConfiguration(createBucketConfiguration);
    }

    Aws::S3::Model::CreateBucketOutcome outcome =
    client.CreateBucket(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: createBucket: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
        std::endl;
        return false;
    } else {
        std::cout << "Created the bucket, '" << bucketName <<
            "', in the region, '" << clientConfig.region << "'." <<
        std::endl;
    }
}

// 2. Upload a local file to the bucket.
Aws::String key = "key-for-test";
{
    Aws::S3::Model::PutObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(key);

    std::shared_ptr<Aws::FStream> input_data =
        Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
            uploadFilePath,
            std::ios_base::in |
            std::ios_base::binary);

    if (!input_data->is_open()) {
```

```
std::cerr << "Error: unable to open file, '" << uploadFilePath <<
    ".\"
    << std::endl;
    AwsDoc::S3::deleteBucket(bucketName, client);
    return false;
}

request.SetBody(input_data);

Aws::S3::Model::PutObjectOutcome outcome =
    client.PutObject(request);

if (!outcome.IsSuccess()) {
    std::cerr << "Error: putObject: " <<
        outcome.GetError().GetMessage() << std::endl;
    AwsDoc::S3::deleteObjectFromBucket(bucketName, key, client);
    AwsDoc::S3::deleteBucket(bucketName, client);
    return false;
} else {
    std::cout << "Added the object with the key, '" << key
        << "', to the bucket, '"
        << bucketName << "'.\" << std::endl;
}
}

// 3. Download the object to a local file.
{
    Aws::S3::Model::GetObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(key);

    Aws::S3::Model::GetObjectOutcome outcome =
        client.GetObject(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: getObject: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        std::cout << "Downloaded the object with the key, '" << key
            << "', in the bucket, '"
            << bucketName << "'.\" << std::endl;
    }
}
```

```
        Aws::IOStream &ioStream = outcome.GetResultWithOwnership().
            GetBody();
        Aws::OFStream outputStream(saveFilePath,
                                   std::ios_base::out | std::ios_base::binary);
        if (!outStream.is_open()) {
            std::cout << "Error: unable to open file, '" << saveFilePath <<
                "'." << std::endl;
        } else {
            outputStream << ioStream.rdbuf();
            std::cout << "Wrote the downloaded object to the file '"
                << saveFilePath << "'." << std::endl;
        }
    }
}

// 4. Copy the object to a different "folder" in the bucket.
Aws::String copiedToKey = "test-folder/" + key;
{
    Aws::S3::Model::CopyObjectRequest request;
    request.WithBucket(bucketName)
        .WithKey(copiedToKey)
        .WithCopySource(bucketName + "/" + key);

    Aws::S3::Model::CopyObjectOutcome outcome =
        client.CopyObject(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error: copyObject: " <<
            outcome.GetError().GetMessage() << std::endl;
    } else {
        std::cout << "Copied the object with the key, '" << key
            << "', to the key, '" << copiedToKey
            << "', in the bucket, '" << bucketName << "'." << std::endl;
    }
}

// 5. List objects in the bucket.
{
    Aws::S3::Model::ListObjectsV2Request request;
    request.WithBucket(bucketName);

    Aws::String continuationToken;
    Aws::Vector<Aws::S3::Model::Object> allObjects;
```

```
do {
    if (!continuationToken.empty()) {
        request.SetContinuationToken(continuationToken);
    }
    Aws::S3::Model::ListObjectsV2Outcome outcome = client.ListObjectsV2(
        request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: ListObjects: " <<
            outcome.GetError().GetMessage() << std::endl;
        break;
    } else {
        Aws::Vector<Aws::S3::Model::Object> objects =
            outcome.GetResult().GetContents();
        allObjects.insert(allObjects.end(), objects.begin(),
objects.end());
        continuationToken = outcome.GetResult().GetContinuationToken();
    }
} while (!continuationToken.empty());

std::cout << allObjects.size() << " objects in the bucket, " <<
bucketName
    << ":" << std::endl;

for (Aws::S3::Model::Object &object: allObjects) {
    std::cout << "    " << object.GetKey() << "" << std::endl;
}
}

// 6. Delete all objects in the bucket.
// All objects in the bucket must be deleted before deleting the bucket.
AwsDoc::S3::deleteObjectFromBucket(bucketName, copiedToKey, client);
AwsDoc::S3::deleteObjectFromBucket(bucketName, key, client);

// 7. Delete the bucket.
return AwsDoc::S3::deleteBucket(bucketName, client);
}

bool AwsDoc::S3::deleteObjectFromBucket(const Aws::String &bucketName,
                                        const Aws::String &key,
                                        Aws::S3::S3Client &client) {
    Aws::S3::Model::DeleteObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(key);
}
```



```

    Aws::S3::Model::DeleteObjectOutcome outcome =
        client.DeleteObject(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: deleteObject: " <<
            outcome.GetError().GetMessage() << std::endl;
    } else {
        std::cout << "Deleted the object with the key, '" << key
            << "', from the bucket, '"
            << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

bool
AwsDoc::S3::deleteBucket(const Aws::String &bucketName, Aws::S3::S3Client
    &client) {
    Aws::S3::Model::DeleteBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketOutcome outcome =
        client.DeleteBucket(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: deleteBucket: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        std::cout << "Deleted the bucket, '" << bucketName << "'." << std::endl;
    }
    return outcome.IsSuccess();
}


```

- APIEinzelheiten finden Sie in den folgenden Themen unter AWS SDK for C++ APIReferenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)

- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Definieren Sie eine Struktur, die die vom Szenario verwendete Bucket- und Objektaktionen umschließt.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// ListBuckets lists the buckets in the current account.
func (basics BucketBasics) ListBuckets() ([]types.Bucket, error) {
    result, err := basics.S3Client.ListBuckets(context.TODO(),
        &s3.ListBucketsInput{})
    var buckets []types.Bucket
    if err != nil {
        log.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
    } else {
        buckets = result.Buckets
    }
    return buckets, err
}
```

```
}

// BucketExists checks whether a bucket exists in the current account.
func (basics BucketBasics) BucketExists(bucketName string) (bool, error) {
    _, err := basics.S3Client.HeadBucket(context.TODO(), &s3.HeadBucketInput{
        Bucket: aws.String(bucketName),
    })
    exists := true
    if err != nil {
        var apiError smithy.APIError
        if errors.As(err, &apiError) {
            switch apiError.(type) {
            case *types.NotFound:
                log.Printf("Bucket %v is available.\n", bucketName)
                exists = false
                err = nil
            default:
                log.Printf("Either you don't have access to bucket %v or another error
occurred. "+
                    "Here's what happened: %v\n", bucketName, err)
            }
        }
    } else {
        log.Printf("Bucket %v exists and you already own it.", bucketName)
    }

    return exists, err
}

// CreateBucket creates a bucket with the specified name in the specified Region.
func (basics BucketBasics) CreateBucket(name string, region string) error {
    _, err := basics.S3Client.CreateBucket(context.TODO(), &s3.CreateBucketInput{
        Bucket: aws.String(name),
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            LocationConstraint: types.BucketLocationConstraint(region),
        },
    })
    if err != nil {
        log.Printf("Couldn't create bucket %v in Region %v. Here's why: %v\n",
            name, region, err)
    }
}
```

```
}
return err
}

// UploadFile reads from a file and puts the data into an object in a bucket.
func (basics BucketBasics) UploadFile(bucketName string, objectKey string,
    fileName string) error {
    file, err := os.Open(fileName)
    if err != nil {
        log.Printf("Couldn't open file %v to upload. Here's why: %v\n", fileName, err)
    } else {
        defer file.Close()
        _, err = basics.S3Client.PutObject(context.TODO(), &s3.PutObjectInput{
            Bucket: aws.String(bucketName),
            Key:     aws.String(objectKey),
            Body:    file,
        })
        if err != nil {
            log.Printf("Couldn't upload file %v to %v:%v. Here's why: %v\n",
                fileName, bucketName, objectKey, err)
        }
    }
    return err
}

// UploadLargeObject uses an upload manager to upload data to an object in a
// bucket.
// The upload manager breaks large data into parts and uploads the parts
// concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
    largeObject []byte) error {
    largeBuffer := bytes.NewReader(largeObject)
    var partMiBs int64 = 10
    uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
        u.PartSize = partMiBs * 1024 * 1024
    })
    _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
        Bucket: aws.String(bucketName),
        Key:     aws.String(objectKey),
        Body:    largeBuffer,
    })
}
```

```
    })
    if err != nil {
        log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }

    return err
}

// DownloadFile gets an object from a bucket and stores it in a local file.
func (basics BucketBasics) DownloadFile(bucketName string, objectKey string,
    fileName string) error {
    result, err := basics.S3Client.GetObject(context.TODO(), &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't get object %v:%v. Here's why: %v\n", bucketName,
            objectKey, err)
        return err
    }
    defer result.Body.Close()
    file, err := os.Create(fileName)
    if err != nil {
        log.Printf("Couldn't create file %v. Here's why: %v\n", fileName, err)
        return err
    }
    defer file.Close()
    body, err := io.ReadAll(result.Body)
    if err != nil {
        log.Printf("Couldn't read object body from %v. Here's why: %v\n", objectKey,
            err)
    }
    _, err = file.Write(body)
    return err
}

// DownloadLargeObject uses a download manager to download an object from a
bucket.
```

```
// The download manager gets the data in parts and writes them to a buffer until
// all of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey
string) ([]byte, error) {
    var partMiBs int64 = 10
    downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader)
    {
        d.PartSize = partMiBs * 1024 * 1024
    })
    buffer := manager.NewWriteAtBuffer([]byte{})
    _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:     aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return buffer.Bytes(), err
}

// CopyToFolder copies an object in a bucket to a subfolder in the same bucket.
func (basics BucketBasics) CopyToFolder(bucketName string, objectKey string,
folderName string) error {
    _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
        Bucket:      aws.String(bucketName),
        CopySource:  aws.String(fmt.Sprintf("%v/%v", bucketName, objectKey)),
        Key:         aws.String(fmt.Sprintf("%v/%v", folderName, objectKey)),
    })
    if err != nil {
        log.Printf("Couldn't copy object from %v:%v to %v:%v/%v. Here's why: %v\n",
            bucketName, objectKey, bucketName, folderName, objectKey, err)
    }
    return err
}

// CopyToBucket copies an object in a bucket to another bucket.
func (basics BucketBasics) CopyToBucket(sourceBucket string, destinationBucket
string, objectKey string) error {
```

```
_, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
    Bucket:      aws.String(destinationBucket),
    CopySource:  aws.String(fmt.Sprintf("%v/%v", sourceBucket, objectKey)),
    Key:         aws.String(objectKey),
})
if err != nil {
    log.Printf("Couldn't copy object from %v:%v to %v:%v. Here's why: %v\n",
        sourceBucket, objectKey, destinationBucket, objectKey, err)
}
return err
}

// ListObjects lists the objects in a bucket.
func (basics BucketBasics) ListObjects(bucketName string) ([]types.Object, error)
{
    result, err := basics.S3Client.ListObjectsV2(context.TODO(),
        &s3.ListObjectsV2Input{
            Bucket: aws.String(bucketName),
        })
    var contents []types.Object
    if err != nil {
        log.Printf("Couldn't list objects in bucket %v. Here's why: %v\n", bucketName,
            err)
    } else {
        contents = result.Contents
    }
    return contents, err
}

// DeleteObjects deletes a list of objects from a bucket.
func (basics BucketBasics) DeleteObjects(bucketName string, objectKeys []string)
error {
    var objectIds []types.ObjectIdentifier
    for _, key := range objectKeys {
        objectIds = append(objectIds, types.ObjectIdentifier{Key: aws.String(key)})
    }
    output, err := basics.S3Client.DeleteObjects(context.TODO(),
        &s3.DeleteObjectsInput{
            Bucket: aws.String(bucketName),
            Delete: &types.Delete{Objects: objectIds},
        })
}
```

```
    })
    if err != nil {
        log.Printf("Couldn't delete objects from bucket %v. Here's why: %v\n",
            bucketName, err)
    } else {
        log.Printf("Deleted %v objects.\n", len(output.Deleted))
    }
    return err
}

// DeleteBucket deletes a bucket. The bucket must be empty or an error is
// returned.
func (basics BucketBasics) DeleteBucket(bucketName string) error {
    _, err := basics.S3Client.DeleteBucket(context.TODO(), &s3.DeleteBucketInput{
        Bucket: aws.String(bucketName)})
    if err != nil {
        log.Printf("Couldn't delete bucket %v. Here's why: %v\n", bucketName, err)
    }
    return err
}
```

Führen Sie ein interaktives Szenario aus, das Ihnen zeigt, wie Sie mit S3-Buckets und Objekten arbeiten.

```
// RunGetStartedScenario is an interactive example that shows you how to use
// Amazon
// Simple Storage Service (Amazon S3) to create an S3 bucket and use it to store
// objects.
//
// 1. Create a bucket.
// 2. Upload a local file to the bucket.
// 3. Upload a large object to the bucket by using an upload manager.
// 4. Download an object to a local file.
// 5. Download a large object by using a download manager.
// 6. Copy an object to a different folder in the bucket.
// 7. List objects in the bucket.
// 8. Delete all objects in the bucket.
// 9. Delete the bucket.
```



```
//
// This example creates an Amazon S3 service client from the specified sdkConfig
// so that
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
// example.
// This package can be found in the ..\..\demotools folder of this repo.
func RunGetStartedScenario(sdkConfig aws.Config, questioner
demotools.IQuestioner) {
defer func() {
if r := recover(); r != nil {
fmt.Println("Something went wrong with the demo.\n", r)
}
}()

log.Println(strings.Repeat("-", 88))
log.Println("Welcome to the Amazon S3 getting started demo.")
log.Println(strings.Repeat("-", 88))

s3Client := s3.NewFromConfig(sdkConfig)
bucketBasics := actions.BucketBasics{S3Client: s3Client}

count := 10
log.Printf("Let's list up to %v buckets for your account:", count)
buckets, err := bucketBasics.ListBuckets()
if err != nil {
panic(err)
}
if len(buckets) == 0 {
log.Println("You don't have any buckets!")
} else {
if count > len(buckets) {
count = len(buckets)
}
for _, bucket := range buckets[:count] {
log.Printf("\t\t%v\n", *bucket.Name)
}
}

bucketName := questioner.Ask("Let's create a bucket. Enter a name for your
bucket:",
demotools.NotEmpty{})
bucketExists, err := bucketBasics.BucketExists(bucketName)
```

```
if err != nil {
    panic(err)
}
if !bucketExists {
    err = bucketBasics.CreateBucket(bucketName, sdkConfig.Region)
    if err != nil {
        panic(err)
    } else {
        log.Println("Bucket created.")
    }
}
log.Println(strings.Repeat("-", 88))

fmt.Println("Let's upload a file to your bucket.")
smallFile := questioner.Ask("Enter the path to a file you want to upload:",
    demotools.NotEmpty{})
const smallKey = "doc-example-key"
err = bucketBasics.UploadFile(bucketName, smallKey, smallFile)
if err != nil {
    panic(err)
}
log.Printf("Uploaded %v as %v.\n", smallFile, smallKey)
log.Println(strings.Repeat("-", 88))

mibs := 30
log.Printf("Let's create a slice of %v MiB of random bytes and upload it to your
bucket. ", mibs)
questioner.Ask("Press Enter when you're ready.")
largeBytes := make([]byte, 1024*1024*mibs)
_, _ = rand.Read(largeBytes)
largeKey := "doc-example-large"
log.Println("Uploading...")
err = bucketBasics.UploadLargeObject(bucketName, largeKey, largeBytes)
if err != nil {
    panic(err)
}
log.Printf("Uploaded %v MiB object as %v", mibs, largeKey)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's download %v to a file.", smallKey)
downloadFileName := questioner.Ask("Enter a name for the downloaded file:",
    demotools.NotEmpty{})
err = bucketBasics.DownloadFile(bucketName, smallKey, downloadFileName)
if err != nil {
```

```
panic(err)
}
log.Printf("File %v downloaded.", downloadFileName)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's download the %v MiB object.", mibs)
questioner.Ask("Press Enter when you're ready.")
log.Println("Downloading...")
largeDownload, err := bucketBasics.DownloadLargeObject(bucketName, largeKey)
if err != nil {
    panic(err)
}
log.Printf("Downloaded %v bytes.", len(largeDownload))
log.Println(strings.Repeat("-", 88))

log.Printf("Let's copy %v to a folder in the same bucket.", smallKey)
folderName := questioner.Ask("Enter a folder name: ", demotools.NotEmpty{})
err = bucketBasics.CopyToFolder(bucketName, smallKey, folderName)
if err != nil {
    panic(err)
}
log.Printf("Copied %v to %v/%v.\n", smallKey, folderName, smallKey)
log.Println(strings.Repeat("-", 88))

log.Println("Let's list the objects in your bucket.")
questioner.Ask("Press Enter when you're ready.")
objects, err := bucketBasics.ListObjects(bucketName)
if err != nil {
    panic(err)
}
log.Printf("Found %v objects.\n", len(objects))
var objKeys []string
for _, object := range objects {
    objKeys = append(objKeys, *object.Key)
    log.Printf("\t\t%v\n", *object.Key)
}
log.Println(strings.Repeat("-", 88))

if questioner.AskBool("Do you want to delete your bucket and all of its "+
    "contents? (y/n)", "y") {
    log.Println("Deleting objects.")
    err = bucketBasics.DeleteObjects(bucketName, objKeys)
    if err != nil {
        panic(err)
    }
}
```

```
}
log.Println("Deleting bucket.")
err = bucketBasics.DeleteBucket(bucketName)
if err != nil {
    panic(err)
}
log.Printf("Deleting downloaded file %v.\n", downloadFileName)
err = os.Remove(downloadFileName)
if err != nil {
    panic(err)
}
} else {
    log.Println("Okay. Don't forget to delete objects from your bucket to avoid
charges.")
}
log.Println(strings.Repeat("-", 88))

log.Println("Thanks for watching!")
log.Println(strings.Repeat("-", 88))
}
```

- API Einzelheiten finden Sie in den folgenden Themen unter [AWS SDK for Go API Referenz](#).
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Ein Szenariobeispiel.

```
import java.io.IOException;
import java.util.Scanner;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 *
 * This Java code example performs the following tasks:
 *
 * 1. Creates an Amazon S3 bucket.
 * 2. Uploads an object to the bucket.
 * 3. Downloads the object to another local file.
 * 4. Uploads an object using multipart upload.
 * 5. List all objects located in the Amazon S3 bucket.
 * 6. Copies the object to another Amazon S3 bucket.
 * 7. Deletes the object from the Amazon S3 bucket.
 * 8. Deletes the Amazon S3 bucket.
 */
```

```
public class S3Scenario {

    public static Scanner scanner = new Scanner(System.in);
    static S3Actions s3Actions = new S3Actions();
    public static final String DASHES = new String(new char[80]).replace("\0",
"-");
    private static final Logger logger =
LoggerFactory.getLogger(S3Scenario.class);
    public static void main(String[] args) throws IOException {
        final String usage = ""
            Usage:
                <key> <objectPath> <savePath> <toBucket>

            Where:
                key - The unique identifier for the object stored in the S3
bucket.
                objectPath - The full file path of the object within the S3
bucket (e.g., "documents/reports/annual_report.pdf").
                savePath - The local file path where the object will be
downloaded and saved (e.g., "C:/Users/username/Downloads/annual_report.pdf").
                toBucket - The name of the S3 bucket to which the object will be
copied.
            """;

        if (args.length != 4) {
            logger.info(usage);
            return;
        }

        String bucketName = "scenario-" + UUID.randomUUID();
        String key = args[0];
        String objectPath = args[1];
        String savePath = args[2];
        String toBucket = args[3];

        logger.info(DASHES);
        logger.info("Welcome to the Amazon Simple Storage Service (S3) example
scenario.");
        logger.info("""
            Amazon S3 is a highly scalable and durable object storage
            service provided by Amazon Web Services (AWS). It is designed to
            store and retrieve
            any amount of data, from anywhere on the web, at any time.
        """);
    }
}
```

The `S3AsyncClient` interface in the AWS SDK for Java 2.x provides a set of methods to programmatically interact with the Amazon S3 (Simple Storage Service) service. This allows developers to automate the management and manipulation of S3 buckets and objects as part of their application deployment pipelines. With S3, teams can focus on building and deploying their applications without having to worry about the underlying storage infrastructure required to host and manage large amounts of data.

This scenario walks you through how to perform key operations for this service.

Let's get started...

```
""");
waitForInputToContinue(scanner);
logger.info(DASHES);

try {
    // Run the methods that belong to this scenario.
    runScenario(bucketName, key, objectPath, savePath, toBucket);

} catch (Throwable rt) {
    Throwable cause = rt.getCause();
    if (cause instanceof S3Exception kmsEx) {
        logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
    } else {
        logger.info("An unexpected error occurred: " + rt.getMessage());
    }
}

private static void runScenario(String bucketName, String key, String
objectPath, String savePath, String toBucket) throws Throwable {
    logger.info(DASHES);
    logger.info("1. Create an Amazon S3 bucket.");
    try {
        CompletableFuture<Void> future =
s3Actions.createBucketAsync(bucketName);
        future.join();
        waitForInputToContinue(scanner);
    }
```

```
    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof S3Exception s3Ex) {
            logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("2. Upload a local file to the Amazon S3 bucket.");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<PutObjectResponse> future =
s3Actions.uploadLocalFileAsync(bucketName, key, objectPath);
        future.join();
        logger.info("File uploaded successfully to {}/{}", bucketName, key);
    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof S3Exception s3Ex) {
            logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("3. Download the object to another local file.");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Void> future =
s3Actions.getObjectBytesAsync(bucketName, key, savePath);
        future.join();
    }
```



```
        logger.info("Successfully obtained bytes from S3 object and wrote to
file {}", savePath);

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof S3Exception s3Ex) {
            logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("4. Perform a multipart upload.");
    waitForInputToContinue(scanner);
    String multipartKey = "multiPartKey";
    try {
        // Call the multipartUpload method
        CompletableFuture<Void> future =
s3Actions.multipartUpload(bucketName, multipartKey);
        future.join();
        logger.info("Multipart upload completed successfully for bucket '{}
and key '{}'", bucketName, multipartKey);

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof S3Exception s3Ex) {
            logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("5. List all objects located in the Amazon S3 bucket.");
    waitForInputToContinue(scanner);
```

```
    try {
        CompletableFuture<Void> future =
s3Actions.listAllObjectsAsync(bucketName);
        future.join();
        logger.info("Object listing completed successfully.");

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof S3Exception s3Ex) {
            logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("6. Copy the object to another Amazon S3 bucket.");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<String> future =
s3Actions.copyBucketObjectAsync(bucketName, key, toBucket);
        String result = future.join();
        logger.info("Copy operation result: {}", result);

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof S3Exception s3Ex) {
            logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("7. Delete objects from the Amazon S3 bucket.");
    waitForInputToContinue(scanner);
```

```
        try {
            CompletableFuture<Void> future =
s3Actions.deleteObjectFromBucketAsync(bucketName, key);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof S3Exception s3Ex) {
                logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }
        try {
            CompletableFuture<Void> future =
s3Actions.deleteObjectFromBucketAsync(bucketName, "multiPartKey");
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof S3Exception s3Ex) {
                logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("8. Delete the Amazon S3 bucket.");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
s3Actions.deleteBucketAsync(bucketName);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof S3Exception s3Ex) {
```

```

        logger.info("S3 error occurred: Error message: {}, Error code
{}", s3Ex.getMessage(), s3Ex.awsErrorDetails().errorCode());
    } else {
        logger.info("An unexpected error occurred: " + rt.getMessage());
    }
    throw cause;
}
waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("You successfully completed the Amazon S3 scenario.");
logger.info(DASHES);
}

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        logger.info("");
        logger.info("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            logger.info("Continuing with the program...");
            logger.info("");
            break;
        } else {
            // Handle invalid input.
            logger.info("Invalid input. Please try again.");
        }
    }
}
}
}
}

```

Eine Wrapper-Klasse, die die Operationen enthält.

```

public class S3Actions {

    private static final Logger logger =
LoggerFactory.getLogger(S3Actions.class);
    private static S3AsyncClient s3AsyncClient;
    public static S3AsyncClient getAsyncClient() {
        if (s3AsyncClient == null) {

```

```
    /*
    The `NettyNioAsyncHttpClient` class is part of the AWS SDK for Java,
    version 2,
    and it is designed to provide a high-performance, asynchronous HTTP
    client for interacting with AWS services.
    It uses the Netty framework to handle the underlying network
    communication and the Java NIO API to
    provide a non-blocking, event-driven approach to HTTP requests and
    responses.
    */

    SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
        .maxConcurrency(50) // Adjust as needed.
        .connectionTimeout(Duration.ofSeconds(60)) // Set the connection
    timeout.

        .readTimeout(Duration.ofSeconds(60)) // Set the read timeout.
        .writeTimeout(Duration.ofSeconds(60)) // Set the write timeout.
        .build();

    ClientOverrideConfiguration overrideConfig =
    ClientOverrideConfiguration.builder()
        .apiCallTimeout(Duration.ofMinutes(2)) // Set the overall API
    call timeout.

        .apiCallAttemptTimeout(Duration.ofSeconds(90)) // Set the
    individual call attempt timeout.
        .build();

    s3AsyncClient = S3AsyncClient.builder()
        .region(Region.US_EAST_1)
        .httpClient(httpClient)
        .overrideConfiguration(overrideConfig)
        .build();
    }
    return s3AsyncClient;
}

/**
 * Creates an S3 bucket asynchronously.
 *
 * @param bucketName the name of the S3 bucket to create
 * @return a {@link CompletableFuture} that completes when the bucket is
    created and ready
 * @throws RuntimeException if there is a failure while creating the bucket
 */
```

```

public CompletableFuture<Void> createBucketAsync(String bucketName) {
    CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
        .bucket(bucketName)
        .build();

    CompletableFuture<CreateBucketResponse> response =
getAsyncClient().createBucket(bucketRequest);
    return response.thenCompose(resp -> {
        S3AsyncWaiter s3Waiter = getAsyncClient().waiter();
        HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
            .bucket(bucketName)
            .build();

        CompletableFuture<WaiterResponse<HeadBucketResponse>>
waiterResponseFuture =
            s3Waiter.waitUntilBucketExists(bucketRequestWait);
        return waiterResponseFuture.thenAccept(waiterResponse -> {
            waiterResponse.matched().response().ifPresent(headBucketResponse
-> {
                logger.info(bucketName + " is ready");
            });
        });
    }).whenComplete((resp, ex) -> {
        if (ex != null) {
            throw new RuntimeException("Failed to create bucket", ex);
        }
    });
}

/**
 * Uploads a local file to an AWS S3 bucket asynchronously.
 *
 * @param bucketName the name of the S3 bucket to upload the file to
 * @param key the key (object name) to use for the uploaded file
 * @param objectPath the local file path of the file to be uploaded
 * @return a {@link CompletableFuture} that completes with the {@link
PutObjectResponse} when the upload is successful, or throws a {@link
RuntimeException} if the upload fails
 */
public CompletableFuture<PutObjectResponse> uploadLocalFileAsync(String
bucketName, String key, String objectPath) {
    PutObjectRequest objectRequest = PutObjectRequest.builder()
        .bucket(bucketName)
        .key(key)

```

```
        .build();

        CompletableFuture<PutObjectResponse> response =
getAsyncClient().putObject(objectRequest,
AsyncRequestBody.fromFile(Paths.get(objectPath)));
        return response.whenComplete((resp, ex) -> {
            if (ex != null) {
                throw new RuntimeException("Failed to upload file", ex);
            }
        });
    }

    /**
     * Asynchronously retrieves the bytes of an object from an Amazon S3 bucket
     * and writes them to a local file.
     *
     * @param bucketName the name of the S3 bucket containing the object
     * @param keyName the key (or name) of the S3 object to retrieve
     * @param path the local file path where the object's bytes will be written
     * @return a {@link CompletableFuture} that completes when the object bytes
     * have been written to the local file
     */
    public CompletableFuture<Void> getObjectBytesAsync(String bucketName, String
keyName, String path) {
        GetObjectRequest objectRequest = GetObjectRequest.builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        CompletableFuture<ResponseBytes<GetObjectResponse>> response =
getAsyncClient().getObject(objectRequest, AsyncResponseTransformer.toBytes());
        return response.thenAccept(objectBytes -> {
            try {
                byte[] data = objectBytes.asByteArray();
                Path filePath = Paths.get(path);
                Files.write(filePath, data);
                logger.info("Successfully obtained bytes from an S3 object");
            } catch (IOException ex) {
                throw new RuntimeException("Failed to write data to file", ex);
            }
        }).whenComplete((resp, ex) -> {
            if (ex != null) {
                throw new RuntimeException("Failed to get object bytes from S3",
ex);
            }
        });
    }
}
```

```
    }
    });
}

/**
 * Asynchronously lists all objects in the specified S3 bucket.
 *
 * @param bucketName the name of the S3 bucket to list objects for
 * @return a {@link CompletableFuture} that completes when all objects have
 * been listed
 */
public CompletableFuture<Void> listAllObjectsAsync(String bucketName) {
    ListObjectsV2Request initialRequest = ListObjectsV2Request.builder()
        .bucket(bucketName)
        .maxKeys(1)
        .build();

    ListObjectsV2Publisher paginator =
getAsyncClient().listObjectsV2Paginator(initialRequest);
    return paginator.subscribe(response -> {
        response.contents().forEach(s3Object -> {
            logger.info("Object key: " + s3Object.key());
        });
    }).thenRun(() -> {
        logger.info("Successfully listed all objects in the bucket: " +
bucketName);
    }).exceptionally(ex -> {
        throw new RuntimeException("Failed to list objects", ex);
    });
}

/**
 * Asynchronously copies an object from one S3 bucket to another.
 *
 * @param fromBucket the name of the source S3 bucket
 * @param objectKey the key (name) of the object to be copied
 * @param toBucket the name of the destination S3 bucket
 * @return a {@link CompletableFuture} that completes with the copy result as
 * a {@link String}
 * @throws RuntimeException if the URL could not be encoded or an S3
 * exception occurred during the copy
 */
public CompletableFuture<String> copyBucketObjectAsync(String fromBucket,
String objectKey, String toBucket) {
```



```
CopyObjectRequest copyReq = CopyObjectRequest.builder()
    .sourceBucket(fromBucket)
    .sourceKey(objectKey)
    .destinationBucket(toBucket)
    .destinationKey(objectKey)
    .build();

CompletableFuture<CopyObjectResponse> response =
getAsyncClient().copyObject(copyReq);
response.whenComplete((copyRes, ex) -> {
    if (copyRes != null) {
        logger.info("The " + objectKey + " was copied to " + toBucket);
    } else {
        throw new RuntimeException("An S3 exception occurred during
copy", ex);
    }
});

return response.thenApply(CopyObjectResponse::copyObjectResult)
    .thenApply(Object::toString);
}

/**
 * Performs a multipart upload to an Amazon S3 bucket.
 *
 * @param bucketName the name of the S3 bucket to upload the file to
 * @param key the key (name) of the file to be uploaded
 * @return a {@link CompletableFuture} that completes when the multipart
upload is successful
 */
public CompletableFuture<Void> multipartUpload(String bucketName, String key)
{
    int mB = 1024 * 1024;

    CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
    .bucket(bucketName)
    .key(key)
    .build();

    return
getAsyncClient().createMultipartUpload(createMultipartUploadRequest)
    .thenCompose(createResponse -> {
        String uploadId = createResponse.uploadId();
```

```
        System.out.println("Upload ID: " + uploadId);

        // Upload part 1.
        UploadPartRequest uploadPartRequest1 =
UploadPartRequest.builder()
        .bucket(bucketName)
        .key(key)
        .uploadId(uploadId)
        .partNumber(1)
        .contentLength((long) (5 * mB)) // Specify the content length
        .build();

        CompletableFuture<CompletedPart> part1Future =
getAsyncClient().uploadPart(uploadPartRequest1,
        AsyncRequestBody.fromByteBuffer(getRandomByteBuffer(5 *
mB)))

        .thenApply(uploadPartResponse -> CompletedPart.builder()
        .partNumber(1)
        .eTag(uploadPartResponse.eTag())
        .build());

        // Upload part 2.
        UploadPartRequest uploadPartRequest2 =
UploadPartRequest.builder()
        .bucket(bucketName)
        .key(key)
        .uploadId(uploadId)
        .partNumber(2)
        .contentLength((long) (3 * mB))
        .build();

        CompletableFuture<CompletedPart> part2Future =
getAsyncClient().uploadPart(uploadPartRequest2,
        AsyncRequestBody.fromByteBuffer(getRandomByteBuffer(3 *
mB)))

        .thenApply(uploadPartResponse -> CompletedPart.builder()
        .partNumber(2)
        .eTag(uploadPartResponse.eTag())
        .build());

        // Combine the results of both parts.
        return CompletableFuture.allOf(part1Future, part2Future)
        .thenCompose(v -> {
            CompletedPart part1 = part1Future.join();
```

```
        CompletedPart part2 = part2Future.join();

        CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
            .parts(part1, part2)
            .build();

        CompleteMultipartUploadRequest
completeMultipartUploadRequest = CompleteMultipartUploadRequest.builder()
            .bucket(bucketName)
            .key(key)
            .uploadId(uploadId)
            .multipartUpload(completedMultipartUpload)
            .build();

        // Complete the multipart upload
        return
getAsyncClient().completeMultipartUpload(completeMultipartUploadRequest);
    });
    })
    .thenAccept(response -> System.out.println("Multipart upload
completed successfully"))
    .exceptionally(ex -> {
        System.err.println("Failed to complete multipart upload: " +
ex.getMessage());
        throw new RuntimeException(ex);
    });
}

/**
 * Deletes an object from an S3 bucket asynchronously.
 *
 * @param bucketName the name of the S3 bucket
 * @param key the key (file name) of the object to be deleted
 * @return a {@link CompletableFuture} that completes when the object has
been deleted
 */
public CompletableFuture<Void> deleteObjectFromBucketAsync(String bucketName,
String key) {
    DeleteObjectRequest deleteObjectRequest = DeleteObjectRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();
```

```
        CompletableFuture<DeleteObjectResponse> response =
getAsyncClient().deleteObject(deleteObjectRequest);
        response.whenComplete((deleteRes, ex) -> {
            if (deleteRes != null) {
                logger.info(key + " was deleted");
            } else {
                throw new RuntimeException("An S3 exception occurred during
delete", ex);
            }
        });

        return response.thenApply(r -> null);
    }

/**
 * Deletes an S3 bucket asynchronously.
 *
 * @param bucket the name of the bucket to be deleted
 * @return a {@link CompletableFuture} that completes when the bucket
deletion is successful, or throws a {@link RuntimeException}
 *         if an error occurs during the deletion process
 */
    public CompletableFuture<Void> deleteBucketAsync(String bucket) {
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
            .bucket(bucket)
            .build();

        CompletableFuture<DeleteBucketResponse> response =
getAsyncClient().deleteBucket(deleteBucketRequest);
        response.whenComplete((deleteRes, ex) -> {
            if (deleteRes != null) {
                logger.info(bucket + " was deleted.");
            } else {
                throw new RuntimeException("An S3 exception occurred during
bucket deletion", ex);
            }
        });
        return response.thenApply(r -> null);
    }

    private static ByteBuffer getRandomByteBuffer(int size) {
        ByteBuffer buffer = ByteBuffer.allocate(size);
        for (int i = 0; i < size; i++) {
            buffer.put((byte) (Math.random() * 256));
        }
    }
}
```

```
    }
    buffer.flip();
    return buffer;
  }
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK for Java 2.x API-Referenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Importieren Sie zunächst alle erforderlichen Module.

```
// Used to check if currently running file is this file.
import { fileURLToPath } from "url";
import { readdirSync, readFileSync, writeFileSync } from "fs";

// Local helper utils.
import { dirnameFromMetaUrl } from "@aws-doc-sdk-examples/lib/utils/util-fs.js";
import { Prompter } from "@aws-doc-sdk-examples/lib/prompter.js";
import { wrapText } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

import {
```

```

S3Client,
CreateBucketCommand,
PutObjectCommand,
ListObjectsCommand,
CopyObjectCommand,
GetObjectCommand,
DeleteObjectsCommand,
DeleteBucketCommand,
} from "@aws-sdk/client-s3";

```

Die vorherigen Importe verweisen auf einige hilfreiche Dienstprogramme. Diese Dienstprogramme befinden sich lokal in dem GitHub Repository, auf das am Anfang dieses Abschnitts verwiesen wird. Zu Ihrer Information finden Sie im Folgenden Implementierungen dieser Dienstprogramme.

```

export const dirnameFromMetaUrl = (metaUrl) =>
  fileURLToPath(new URL(".", metaUrl));

import { select, input, confirm, checkbox } from "@inquirer/prompts";

export class Prompter {
  /**
   * @param {{ message: string, choices: { name: string, value: string }[] }}
  options
  */
  select(options) {
    return select(options);
  }

  /**
   * @param {{ message: string }} options
  */
  input(options) {
    return input(options);
  }

  /**
   * @param {string} prompt
  */
  checkContinue = async (prompt = "") => {
    const prefix = prompt && prompt + " ";
    let ok = await this.confirm({

```

```

    message: `${prefix}Continue?`,
  });
  if (!ok) throw new Error("Exiting...");
};

/**
 * @param {{ message: string }} options
 */
confirm(options) {
  return confirm(options);
}

/**
 * @param {{ message: string, choices: { name: string, value: string }[]}}
options
 */
checkbox(options) {
  return checkbox(options);
}
}

export const wrapText = (text, char = "=") => {
  const rule = char.repeat(80);
  return `${rule}\n  ${text}\n${rule}\n`;
};

```

Objekte in S3 werden in „Buckets“ gespeichert. Definieren wir eine Funktion zum Erstellen eines neuen Buckets.

```

export const createBucket = async () => {
  const bucketName = await prompter.input({
    message: "Enter a bucket name. Bucket names must be globally unique:",
  });
  const command = new CreateBucketCommand({ Bucket: bucketName });
  await s3Client.send(command);
  console.log("Bucket created successfully.\n");
  return bucketName;
};

```

Buckets enthalten „Objekte“. Diese Funktion lädt den Inhalt eines Verzeichnisses als Objekte in Ihren Bucket hoch.

```
export const uploadFilesToBucket = async ({ bucketName, folderPath }) => {
  console.log(`Uploading files from ${folderPath}\n`);
  const keys = readdirSync(folderPath);
  const files = keys.map((key) => {
    const filePath = `${folderPath}/${key}`;
    const fileContent = readFileSync(filePath);
    return {
      Key: key,
      Body: fileContent,
    };
  });

  for (let file of files) {
    await s3Client.send(
      new PutObjectCommand({
        Bucket: bucketName,
        Body: file.Body,
        Key: file.Key,
      }),
    );
    console.log(`${file.Key} uploaded successfully.`);
  }
};
```

Überprüfen Sie nach dem Hochladen von Objekten, ob sie korrekt hochgeladen wurden. Sie können `ListObjects` dafür verwenden. Sie werden die Eigenschaft „Key“ verwenden, die Antwort enthält jedoch noch weitere nützliche Eigenschaften.

```
export const listFilesInBucket = async ({ bucketName }) => {
  const command = new ListObjectsCommand({ Bucket: bucketName });
  const { Contents } = await s3Client.send(command);
  const contentsList = Contents.map((c) => ` • ${c.Key}`).join("\n");
  console.log("\nHere's a list of files in the bucket:");
  console.log(contentsList + "\n");
};
```

Gelegentlich möchten Sie vielleicht ein Objekt von einem Bucket in einen anderen kopieren. Verwenden Sie dafür den `CopyObject` Befehl.

```
export const copyFileFromBucket = async ({ destinationBucket }) => {
```



```
const proceed = await prompter.confirm({
  message: "Would you like to copy an object from another bucket?",
});

if (!proceed) {
  return;
} else {
  const copy = async () => {
    try {
      const sourceBucket = await prompter.input({
        message: "Enter source bucket name:",
      });
      const sourceKey = await prompter.input({
        message: "Enter source key:",
      });
      const destinationKey = await prompter.input({
        message: "Enter destination key:",
      });

      const command = new CopyObjectCommand({
        Bucket: destinationBucket,
        CopySource: `${sourceBucket}/${sourceKey}`,
        Key: destinationKey,
      });
      await s3Client.send(command);
      await copyFileFromBucket({ destinationBucket });
    } catch (err) {
      console.error(`Copy error.`);
      console.error(err);
      const retryAnswer = await prompter.confirm({ message: "Try again?" });
      if (retryAnswer) {
        await copy();
      }
    }
  };
  await copy();
}
};
```

Es gibt keine SDK Methode, um mehrere Objekte aus einem Bucket abzurufen. Stattdessen erstellen Sie eine Liste von Objekten, die Sie herunterladen und durchlaufen können.

```
export const downloadFilesFromBucket = async ({ bucketName }) => {
  const { Contents } = await s3Client.send(
    new ListObjectsCommand({ Bucket: bucketName }),
  );
  const path = await prompter.input({
    message: "Enter destination path for files:",
  });

  for (let content of Contents) {
    const obj = await s3Client.send(
      new GetObjectCommand({ Bucket: bucketName, Key: content.Key }),
    );
    writeFileSync(
      `${path}/${content.Key}`,
      await obj.Body.transformToByteArray(),
    );
  }
  console.log("Files downloaded successfully.\n");
};
```

Es ist Zeit, Ihre Ressourcen zu bereinigen. Ein Bucket muss leer sein, bevor er gelöscht werden kann. Mit diesen beiden Funktionen leeren und löschen Sie den Bucket.

```
export const emptyBucket = async ({ bucketName }) => {
  const listObjectsCommand = new ListObjectsCommand({ Bucket: bucketName });
  const { Contents } = await s3Client.send(listObjectsCommand);
  const keys = Contents.map((c) => c.Key);

  const deleteObjectsCommand = new DeleteObjectsCommand({
    Bucket: bucketName,
    Delete: { Objects: keys.map((key) => ({ Key: key })) },
  });
  await s3Client.send(deleteObjectsCommand);
  console.log(`${bucketName} emptied successfully.\n`);
};

export const deleteBucket = async ({ bucketName }) => {
  const command = new DeleteBucketCommand({ Bucket: bucketName });
  await s3Client.send(command);
  console.log(`${bucketName} deleted successfully.\n`);
};
```

Die Funktion „main“ fasst alles zusammen. Wenn Sie diese Datei direkt ausführen, wird die Funktion „main“ aufgerufen.

```
const main = async () => {
  const OBJECT_DIRECTORY = `${dirnameFromMetaUrl(
    import.meta.url,
  )}../../../../resources/sample_files/.sample_media`;

  try {
    console.log(wrapText("Welcome to the Amazon S3 getting started example."));
    console.log("Let's create a bucket.");
    const bucketName = await createBucket();
    await prompter.confirm({ message: continueMessage });

    console.log(wrapText("File upload."));
    console.log(
      "I have some default files ready to go. You can edit the source code to
      provide your own.",
    );
    await uploadFilesToBucket({
      bucketName,
      folderPath: OBJECT_DIRECTORY,
    });

    await listFilesInBucket({ bucketName });
    await prompter.confirm({ message: continueMessage });

    console.log(wrapText("Copy files."));
    await copyFileFromBucket({ destinationBucket: bucketName });
    await listFilesInBucket({ bucketName });
    await prompter.confirm({ message: continueMessage });

    console.log(wrapText("Download files."));
    await downloadFilesFromBucket({ bucketName });

    console.log(wrapText("Clean up."));
    await emptyBucket({ bucketName });
    await deleteBucket({ bucketName });
  } catch (err) {
    console.error(err);
  }
}
```

```
};
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK for JavaScript API-Referenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <bucketName> <key> <objectPath> <savePath> <toBucket>

    Where:
        bucketName - The Amazon S3 bucket to create.
        key - The key to use.
        objectPath - The path where the file is located (for example, C:/AWS/
book2.pdf).
        savePath - The path where the file is saved after it's downloaded (for
example, C:/AWS/book2.pdf).
        toBucket - An Amazon S3 bucket to where an object is copied to (for
example, C:/AWS/book2.pdf).
    """
}
```

```
if (args.size != 4) {
    println(usage)
    exitProcess(1)
}

val bucketName = args[0]
val key = args[1]
val objectPath = args[2]
val savePath = args[3]
val toBucket = args[4]

// Create an Amazon S3 bucket.
createBucket(bucketName)

// Update a local file to the Amazon S3 bucket.
putObject(bucketName, key, objectPath)

// Download the object to another local file.
getObjectFromMrap(bucketName, key, savePath)

// List all objects located in the Amazon S3 bucket.
listBucketObs(bucketName)

// Copy the object to another Amazon S3 bucket
copyBucketOb(bucketName, key, toBucket)

// Delete the object from the Amazon S3 bucket.
deleteBucketObs(bucketName, key)

// Delete the Amazon S3 bucket.
deleteBucket(bucketName)
println("All Amazon S3 operations were successfully performed")
}

suspend fun createBucket(bucketName: String) {
    val request =
        CreateBucketRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        println("$bucketName is ready")
    }
}
```

```
    }  
  }  
  
suspend fun putObject(  
    bucketName: String,  
    objectKey: String,  
    objectPath: String,  
) {  
    val metadataVal = mutableMapOf<String, String>()  
    metadataVal["myVal"] = "test"  
  
    val request =  
        PutObjectRequest {  
            bucket = bucketName  
            key = objectKey  
            metadata = metadataVal  
            this.body = Paths.get(objectPath).asByteArray()  
        }  
  
    S3Client { region = "us-east-1" }.use { s3 ->  
        val response = s3.putObject(request)  
        println("Tag information is ${response.eTag}")  
    }  
}  
  
suspend fun getObjectFromMrap(  
    bucketName: String,  
    keyName: String,  
    path: String,  
) {  
    val request =  
        GetObjectRequest {  
            key = keyName  
            bucket = bucketName  
        }  
  
    S3Client { region = "us-east-1" }.use { s3 ->  
        s3.getObject(request) { resp ->  
            val myFile = File(path)  
            resp.body?.writeToFile(myFile)  
            println("Successfully read $keyName from $bucketName")  
        }  
    }  
}
```

```
suspend fun listBucketObs(bucketName: String) {
    val request =
        ListObjectsRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->

        val response = s3.listObjects(request)
        response.contents?.forEach { myObject ->
            println("The name of the key is ${myObject.key}")
            println("The owner is ${myObject.owner}")
        }
    }
}

suspend fun copyBucketOb(
    fromBucket: String,
    objectKey: String,
    toBucket: String,
) {
    var encodedUrl = ""
    try {
        encodedUrl = URLEncoder.encode("$fromBucket/$objectKey",
StandardCharsets.UTF_8.toString())
    } catch (e: UnsupportedOperationException) {
        println("URL could not be encoded: " + e.message)
    }

    val request =
        CopyObjectRequest {
            copySource = encodedUrl
            bucket = toBucket
            key = objectKey
        }
    S3Client { region = "us-east-1" }.use { s3 ->
        s3.copyObject(request)
    }
}

suspend fun deleteBucketObs(
    bucketName: String,
    objectName: String,
```

```
) {
    val objectId =
        ObjectIdentifier {
            key = objectName
        }

    val delOb =
        Delete {
            objects = listOf(objectId)
        }

    val request =
        DeleteObjectsRequest {
            bucket = bucketName
            delete = delOb
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteObjects(request)
        println("$objectName was deleted from $bucketName")
    }
}

suspend fun deleteBucket(bucketName: String?) {
    val request =
        DeleteBucketRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteBucket(request)
        println("The $bucketName was successfully deleted!")
    }
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK als API Kotlin-Referenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)

- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

PHP

SDK für PHP

Note

Da ist noch mehr GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
echo("\n");
echo("-----\n");
print("Welcome to the Amazon S3 getting started demo using PHP!\n");
echo("-----\n");

$region = 'us-west-2';

$this->s3client = new S3Client([
    'region' => $region,
]);
/* Inline declaration example
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);
*/

$this->bucketName = "amzn-s3-demo-bucket-" . uniqid();

try {
    $this->s3client->createBucket([
        'Bucket' => $this->bucketName,
        'CreateBucketConfiguration' => ['LocationConstraint' => $region],
    ]);
    echo "Created bucket named: $this->bucketName \n";
} catch (Exception $exception) {
    echo "Failed to create bucket $this->bucketName with error: " .
    $exception->getMessage();
    exit("Please fix error with bucket creation before continuing.");
}
```

```
$fileName = __DIR__ . "/local-file-" . uniqid();
try {
    $this->s3client->putObject([
        'Bucket' => $this->bucketName,
        'Key' => $fileName,
        'SourceFile' => __DIR__ . '/testfile.txt'
    ]);
    echo "Uploaded $fileName to $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to upload $fileName with error: " . $exception-
>getMessage();
    exit("Please fix error with file upload before continuing.");
}

try {
    $file = $this->s3client->getObject([
        'Bucket' => $this->bucketName,
        'Key' => $fileName,
    ]);
    $body = $file->get('Body');
    $body->rewind();
    echo "Downloaded the file and it begins with: {$body->read(26)}.\n";
} catch (Exception $exception) {
    echo "Failed to download $fileName from $this->bucketName with error:
" . $exception->getMessage();
    exit("Please fix error with file downloading before continuing.");
}

try {
    $folder = "copied-folder";
    $this->s3client->copyObject([
        'Bucket' => $this->bucketName,
        'CopySource' => "$this->bucketName/$fileName",
        'Key' => "$folder/$fileName-copy",
    ]);
    echo "Copied $fileName to $folder/$fileName-copy.\n";
} catch (Exception $exception) {
    echo "Failed to copy $fileName with error: " . $exception-
>getMessage();
    exit("Please fix error with object copying before continuing.");
}

try {
```

```
$contents = $this->s3client->listObjectsV2([
    'Bucket' => $this->bucketName,
]);
echo "The contents of your bucket are: \n";
foreach ($contents['Contents'] as $content) {
    echo $content['Key'] . "\n";
}
} catch (Exception $exception) {
    echo "Failed to list objects in $this->bucketName with error: " .
$exception->getMessage();
    exit("Please fix error with listing objects before continuing.");
}

try {
    $objects = [];
    foreach ($contents['Contents'] as $content) {
        $objects[] = [
            'Key' => $content['Key'],
        ];
    }
    $this->s3client->deleteObjects([
        'Bucket' => $this->bucketName,
        'Delete' => [
            'Objects' => $objects,
        ],
    ]);
    $check = $this->s3client->listObjectsV2([
        'Bucket' => $this->bucketName,
    ]);
    if (count($check) <= 0) {
        throw new Exception("Bucket wasn't empty.");
    }
    echo "Deleted all objects and folders from $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to delete $fileName from $this->bucketName with error:
" . $exception->getMessage();
    exit("Please fix error with object deletion before continuing.");
}

try {
    $this->s3client->deleteBucket([
        'Bucket' => $this->bucketName,
    ]);
    echo "Deleted bucket $this->bucketName.\n";
}
```

```
    } catch (Exception $exception) {
        echo "Failed to delete $this->bucketName with error: " . $exception-
>getMessage();
        exit("Please fix error with bucket deletion before continuing.");
    }

    echo "Successfully ran the Amazon S3 with PHP demo.\n";
```

- APIEinzelheiten finden Sie in den folgenden Themen unter AWS SDK for PHP APIReferenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import io
import os
import uuid

import boto3
from boto3.s3.transfer import S3UploadFailedError
from botocore.exceptions import ClientError
```

```
def do_scenario(s3_resource):
    print("-" * 88)
    print("Welcome to the Amazon S3 getting started demo!")
    print("-" * 88)

    bucket_name = f"amzn-s3-demo-bucket-{{uuid.uuid4()}}"
    bucket = s3_resource.Bucket(bucket_name)
    try:
        bucket.create(
            CreateBucketConfiguration={
                "LocationConstraint": s3_resource.meta.client.meta.region_name
            }
        )
        print(f"Created demo bucket named {bucket.name}.")
    except ClientError as err:
        print(f"Tried and failed to create demo bucket {bucket_name}.")
        print(f"\t{err.response['Error']['Code']}: {err.response['Error']
['Message']}")
        print(f"\nCan't continue the demo without a bucket!")
        return

    file_name = None
    while file_name is None:
        file_name = input("\nEnter a file you want to upload to your bucket: ")
        if not os.path.exists(file_name):
            print(f"Couldn't find file {file_name}. Are you sure it exists?")
            file_name = None

    obj = bucket.Object(os.path.basename(file_name))
    try:
        obj.upload_file(file_name)
        print(
            f"Uploaded file {file_name} into bucket {bucket.name} with key
{obj.key}."
        )
    except S3UploadFailedError as err:
        print(f"Couldn't upload file {file_name} to {bucket.name}.")
        print(f"\t{err}")

    answer = input(f"\nDo you want to download {obj.key} into memory (y/n)? ")
    if answer.lower() == "y":
        data = io.BytesIO()
        try:
            obj.download_fileobj(data)
```

```
        data.seek(0)
        print(f"Got your object. Here are the first 20 bytes:\n")
        print(f"\t{data.read(20)}")
    except ClientError as err:
        print(f"Couldn't download {obj.key}.")
        print(
            f"\t{err.response['Error']['Code']}: {err.response['Error']
['Message']}"
        )

    answer = input(
        f"\nDo you want to copy {obj.key} to a subfolder in your bucket (y/n)? "
    )
    if answer.lower() == "y":
        dest_obj = bucket.Object(f"demo-folder/{obj.key}")
        try:
            dest_obj.copy({"Bucket": bucket.name, "Key": obj.key})
            print(f"Copied {obj.key} to {dest_obj.key}.")
        except ClientError as err:
            print(f"Couldn't copy {obj.key} to {dest_obj.key}.")
            print(
                f"\t{err.response['Error']['Code']}: {err.response['Error']
['Message']}"
            )

    print("\nYour bucket contains the following objects:")
    try:
        for o in bucket.objects.all():
            print(f"\t{o.key}")
    except ClientError as err:
        print(f"Couldn't list the objects in bucket {bucket.name}.")
        print(f"\t{err.response['Error']['Code']}: {err.response['Error']
['Message']}"
    )

    answer = input(
        "\nDo you want to delete all of the objects as well as the bucket (y/n)?
"
    )
    if answer.lower() == "y":
        try:
            bucket.objects.delete()
            bucket.delete()
            print(f"Emptied and deleted bucket {bucket.name}.\n")
        except ClientError as err:
```

```
        print(f"Couldn't empty and delete bucket {bucket.name}.")
        print(
            f"\t{err.response['Error']['Code']}:{err.response['Error']
['Message']}"
        )

        print("Thanks for watching!")
        print("-" * 88)

if __name__ == "__main__":
    do_scenario(boto3.resource("s3"))
```

- API Einzelheiten finden Sie in den folgenden Themen unter AWS SDK für Python (Boto3) API-Referenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps the getting started scenario actions.
class ScenarioGettingStarted
```

```
attr_reader :s3_resource

# @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
def initialize(s3_resource)
  @s3_resource = s3_resource
end

# Creates a bucket with a random name in the currently configured account and
# AWS Region.
#
# @return [Aws::S3::Bucket] The newly created bucket.
def create_bucket
  bucket = @s3_resource.create_bucket(
    bucket: "amzn-s3-demo-bucket-#{Random.uuid}",
    create_bucket_configuration: {
      location_constraint: "us-east-1" # Note: only certain regions permitted
    }
  )
  puts("Created demo bucket named #{bucket.name}.")
rescue Aws::Errors::ServiceError => e
  puts("Tried and failed to create demo bucket.")
  puts("\t#{e.code}: #{e.message}")
  puts("\nCan't continue the demo without a bucket!")
  raise
else
  bucket
end

# Requests a file name from the user.
#
# @return The name of the file.
def create_file
  File.open("demo.txt", w) { |f| f.write("This is a demo file.") }
end

# Uploads a file to an Amazon S3 bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket object representing the upload
destination
# @return [Aws::S3::Object] The Amazon S3 object that contains the uploaded
file.
def upload_file(bucket)
  File.open("demo.txt", "w+") { |f| f.write("This is a demo file.") }
  s3_object = bucket.object(File.basename("demo.txt"))
end
```



```
s3_object.upload_file("demo.txt")
puts("Uploaded file demo.txt into bucket #{bucket.name} with key
#{s3_object.key}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't upload file demo.txt to #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  s3_object
end

# Downloads an Amazon S3 object to a file.
#
# @param s3_object [Aws::S3::Object] The object to download.
def download_file(s3_object)
  puts("\nDo you want to download #{s3_object.key} to a local file (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    puts("Enter a name for the downloaded file:")
    file_name = gets.chomp
    s3_object.download_file(file_name)
    puts("Object #{s3_object.key} successfully downloaded to #{file_name}.")
  end
end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't download #{s3_object.key}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

# Copies an Amazon S3 object to a subfolder within the same bucket.
#
# @param source_object [Aws::S3::Object] The source object to copy.
# @return [Aws::S3::Object, nil] The destination object.
def copy_object(source_object)
  dest_object = nil
  puts("\nDo you want to copy #{source_object.key} to a subfolder in your
bucket (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    dest_object = source_object.bucket.object("demo-folder/
#{source_object.key}")
    dest_object.copy_from(source_object)
    puts("Copied #{source_object.key} to #{dest_object.key}.")
  end
end
```

```
rescue Aws::Errors::ServiceError => e
  puts("Couldn't copy #{source_object.key}.")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  dest_object
end

# Lists the objects in an Amazon S3 bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to query.
def list_objects(bucket)
  puts("\nYour bucket contains the following objects:")
  bucket.objects.each do |obj|
    puts("\t#{obj.key}")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list the objects in bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?")
  answer = gets.chomp.downcase
  if answer == "y"
    bucket.objects.batch_delete!
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

# Runs the Amazon S3 getting started scenario.
def run_scenario(scenario)
  puts("-" * 88)
```

```
puts("Welcome to the Amazon S3 getting started demo!")
puts("-" * 88)

bucket = scenario.create_bucket
s3_object = scenario.upload_file(bucket)
scenario.download_file(s3_object)
scenario.copy_object(s3_object)
scenario.list_objects(bucket)
scenario.delete_bucket(bucket)

puts("Thanks for watching!")
puts("-" * 88)
rescue Aws::Errors::ServiceError
  puts("Something went wrong with the demo!")
end

run_scenario(ScenarioGettingStarted.new(Aws::S3::Resource.new)) if $PROGRAM_NAME
== __FILE__
```

- APIEinzelheiten finden Sie in den folgenden Themen unter AWS SDK for Ruby APIReferenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Rust

SDKfür Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Code für die binäre Crate-Datei, die das Szenario ausführt.

```
#![allow(clippy::result_large_err)]

//! Purpose
//! Shows how to use the AWS SDK for Rust to get started using
//! Amazon Simple Storage Service (Amazon S3). Create a bucket, move objects
    into and out of it,
//! and delete all resources at the end of the demo.
//!
//! This example follows the steps in "Getting started with Amazon S3" in the
    Amazon S3
    user guide.
    - https://docs.aws.amazon.com/AmazonS3/latest/userguide/GetStartedWithS3.html

use aws_config::meta::region::RegionProviderChain;
use aws_sdk_s3::{config::Region, Client};
use s3_code_examples::error::S3ExampleError;
use uuid::Uuid;

#[tokio::main]
async fn main() -> Result<(), S3ExampleError> {
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));
    let region = region_provider.region().await.unwrap();
    let shared_config =
aws_config::from_env().region(region_provider).load().await;
    let client = Client::new(&shared_config);
    let bucket_name = format!("amzn-s3-demo-bucket-{}", Uuid::new_v4());
    let file_name = "s3/testfile.txt".to_string();
    let key = "test file key name".to_string();
    let target_key = "target_key".to_string();

    if let Err(e) = run_s3_operations(region, client, bucket_name, file_name,
key, target_key).await
    {
        eprintln!("{:?}", e);
    };

    Ok(())
}
```

```

async fn run_s3_operations(
    region: Region,
    client: Client,
    bucket_name: String,
    file_name: String,
    key: String,
    target_key: String,
) -> Result<(), S3ExampleError> {
    s3_code_examples::create_bucket(&client, &bucket_name, &region).await?;
    let run_example: Result<(), S3ExampleError> = (async {
        s3_code_examples::upload_object(&client, &bucket_name, &file_name,
&key).await?;
        let _object = s3_code_examples::download_object(&client, &bucket_name,
&key).await;
        s3_code_examples::copy_object(&client, &bucket_name, &bucket_name, &key,
&target_key)
            .await?;
        s3_code_examples::list_objects(&client, &bucket_name).await?;
        s3_code_examples::clear_bucket(&client, &bucket_name).await?;
        Ok(())
    })
    .await;
    if let Err(err) = run_example {
        eprintln!("Failed to complete getting-started example: {err:?}");
    }
    s3_code_examples::delete_bucket(&client, &bucket_name).await?;

    Ok(())
}

```

Allgemeine Aktionen, die im Szenario verwendet werden.

```

pub async fn create_bucket(
    client: &aws_sdk_s3::Client,
    bucket_name: &str,
    region: &aws_config::Region,
) -> Result<Option<aws_sdk_s3::operation::create_bucket::CreateBucketOutput>,
S3ExampleError> {
    let constraint =
aws_sdk_s3::types::BucketLocationConstraint::from(region.to_string().as_str());
    let cfg = aws_sdk_s3::types::CreateBucketConfiguration::builder()
        .location_constraint(constraint)

```

```
        .build();
let create = client
    .create_bucket()
    .create_bucket_configuration(cfg)
    .bucket(bucket_name)
    .send()
    .await;

// BucketAlreadyExists and BucketAlreadyOwnedByYou are not problems for this
task.
create.map(Some).or_else(|err| {
    if err
        .as_service_error()
        .map(|se| se.is_bucket_already_exists() ||
se.is_bucket_already_owned_by_you())
        == Some(true)
    {
        Ok(None)
    } else {
        Err(S3ExampleError::from(err))
    }
})
}

pub async fn upload_object(
    client: &aws_sdk_s3::Client,
    bucket_name: &str,
    file_name: &str,
    key: &str,
) -> Result<aws_sdk_s3::operation::put_object::PutObjectOutput, S3ExampleError> {
    let body =
aws_sdk_s3::primitives::ByteStream::from_path(std::path::Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await
        .map_err(S3ExampleError::from)
}

pub async fn download_object(
    client: &aws_sdk_s3::Client,
```

```
    bucket_name: &str,
    key: &str,
) -> Result<aws_sdk_s3::operation::get_object::GetObjectOutput, S3ExampleError> {
    client
        .get_object()
        .bucket(bucket_name)
        .key(key)
        .send()
        .await
        .map_err(S3ExampleError::from)
}

/// Copy an object from one bucket to another.
pub async fn copy_object(
    client: &aws_sdk_s3::Client,
    source_bucket: &str,
    destination_bucket: &str,
    source_object: &str,
    destination_object: &str,
) -> Result<(), S3ExampleError> {
    let source_key = format!("{source_bucket}/{source_object}");
    let response = client
        .copy_object()
        .copy_source(&source_key)
        .bucket(destination_bucket)
        .key(destination_object)
        .send()
        .await?;

    println!(
        "Copied from {source_key} to {destination_bucket}/{destination_object}
with etag {}",
        response
            .copy_object_result
            .unwrap_or_else(||
aws_sdk_s3::types::CopyObjectResult::builder().build())
            .e_tag()
            .unwrap_or("missing")
    );
    Ok(())
}

pub async fn list_objects(client: &aws_sdk_s3::Client, bucket: &str) ->
Result<(), S3ExampleError> {
```

```

let mut response = client
    .list_objects_v2()
    .bucket(bucket.to_owned())
    .max_keys(10) // In this example, go 10 at a time.
    .into_paginator()
    .send();

while let Some(result) = response.next().await {
    match result {
        Ok(output) => {
            for object in output.contents() {
                println!(" - {}", object.key().unwrap_or("Unknown"));
            }
        }
        Err(err) => {
            eprintln!("{err:?}")
        }
    }
}

Ok(())
}

/// Given a bucket, remove all objects in the bucket, and then ensure no objects
/// remain in the bucket.
pub async fn clear_bucket(
    client: &aws_sdk_s3::Client,
    bucket_name: &str,
) -> Result<Vec<String>, S3ExampleError> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;

    // delete_objects no longer needs to be mutable.
    let objects_to_delete: Vec<String> = objects
        .contents()
        .iter()
        .filter_map(|obj| obj.key())
        .map(String::from)
        .collect();

    if objects_to_delete.is_empty() {
        return Ok(vec![]);
    }

    let return_keys = objects_to_delete.clone();

```



```
delete_objects(client, bucket_name, objects_to_delete).await?;

let objects = client.list_objects_v2().bucket(bucket_name).send().await?;

eprintln!("{objects:?}");

match objects.key_count {
    Some(0) => Ok(return_keys),
    _ => Err(S3ExampleError::new(
        "There were still objects left in the bucket.",
    )),
}

pub async fn delete_bucket(
    client: &aws_sdk_s3::Client,
    bucket_name: &str,
) -> Result<(), S3ExampleError> {
    let resp = client.delete_bucket().bucket(bucket_name).send().await;
    match resp {
        Ok(_) => Ok(()),
        Err(err) => {
            if err
                .as_service_error()
                .and_then(aws_sdk_s3::error::ProvideErrorMetadata::code)
                == Some("NoSuchBucket")
            {
                Ok(())
            } else {
                Err(S3ExampleError::from(err))
            }
        }
    }
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK als API-Referenz zu Rust.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)

- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).

" Create an Amazon Simple Storage Service (Amazon S3) bucket. "
TRY.
    lo_s3->createbucket(
        iv_bucket = iv_bucket_name
    ).
    MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrdyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrdyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.

"Upload an object to an S3 bucket."
TRY.
    "Get contents of file from application server."
    DATA lv_file_content TYPE xstring.
    OPEN DATASET iv_key FOR INPUT IN BINARY MODE.
    READ DATASET iv_key INTO lv_file_content.
    CLOSE DATASET iv_key.

    lo_s3->putobject(
```

```
        iv_bucket = iv_bucket_name
        iv_key = iv_key
        iv_body = lv_file_content
    ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" Get an object from a bucket. "
TRY.
    DATA(lo_result) = lo_s3->getobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
    ).
    DATA(lv_object_data) = lo_result->get_body( ).
    MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" Copy an object to a subfolder in a bucket. "
TRY.
    lo_s3->copyobject(
        iv_bucket = iv_bucket_name
        iv_key = |{ iv_copy_to_folder }/{ iv_key }|
        iv_copysource = |{ iv_bucket_name }/{ iv_key }|
    ).
    MESSAGE 'Object copied to a subfolder.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" List objects in the bucket. "
TRY.
    DATA(lo_list) = lo_s3->listobjects(
        iv_bucket = iv_bucket_name
    ).
    MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
```

```
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  ENDRY.
  DATA text TYPE string VALUE 'Object List - '.
  DATA lv_object_key TYPE /aws1/s3_objectkey.
  LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
    lv_object_key = lo_object->get_key( ).
    CONCATENATE lv_object_key ', ' INTO text.
  ENDLLOOP.
  MESSAGE text TYPE'I'.

" Delete the objects in a bucket. "
  TRY.
    lo_s3->deleteobject(
      iv_bucket = iv_bucket_name
      iv_key = iv_key
    ).
    lo_s3->deleteobject(
      iv_bucket = iv_bucket_name
      iv_key = |{ iv_copy_to_folder }/{ iv_key }|
    ).
    MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  ENDRY.

" Delete the bucket. "
  TRY.
    lo_s3->deletebucket(
      iv_bucket = iv_bucket_name
    ).
    MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  ENDRY.
```

- APIEinzelheiten finden Sie in den folgenden Themen unter AWS SDKals SAP ABAP API Referenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)

- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3

import Foundation
import AWSS3
import Smithy
import ClientRuntime

/// A class containing all the code that interacts with the AWS SDK for Swift.
public class ServiceHandler {
    let configuration: S3Client.S3ClientConfiguration
    let client: S3Client

    enum HandlerError: Error {
        case getObjectBody(String)
        case readGetObjectBody(String)
        case missingContents(String)
    }

    /// Initialize and return a new ``ServiceHandler`` object, which is used to
    drive the AWS calls
    /// used for the example.
    ///
    /// - Returns: A new ``ServiceHandler`` object, ready to be called to
    ///           execute AWS operations.
    public init() async throws {
```

```
do {
    configuration = try await S3Client.S3ClientConfiguration()
    // configuration.region = "us-east-2" // Uncomment this to set the
region programmatically.
    client = S3Client(config: configuration)
}
catch {
    print("ERROR: ", dump(error, name: "Initializing S3 client"))
    throw error
}
}

/// Create a new user given the specified name.
///
/// - Parameters:
///   - name: Name of the bucket to create.
/// Throws an exception if an error occurs.
public func createBucket(name: String) async throws {
    var input = CreateBucketInput(
        bucket: name
    )

    // For regions other than "us-east-1", you must set the
locationConstraint in the createBucketConfiguration.
    // For more information, see LocationConstraint in the S3 API guide.
    // https://docs.aws.amazon.com/AmazonS3/latest/API/
API_CreateBucket.html#API_CreateBucket_RequestBody
    if let region = configuration.region {
        if region != "us-east-1" {
            input.createBucketConfiguration =
S3ClientTypes.CreateBucketConfiguration(locationConstraint:
S3ClientTypes.BucketLocationConstraint(rawValue: region))
        }
    }

    do {
        _ = try await client.createBucket(input: input)
    }
    catch let error as BucketAlreadyOwnedByYou {
        print("The bucket '\(name)' already exists and is owned by you. You
may wish to ignore this exception.")
        throw error
    }
}
```

```
        catch {
            print("ERROR: ", dump(error, name: "Creating a bucket"))
            throw error
        }
    }

    /// Delete a bucket.
    /// - Parameter name: Name of the bucket to delete.
    public func deleteBucket(name: String) async throws {
        let input = DeleteBucketInput(
            bucket: name
        )
        do {
            _ = try await client.deleteBucket(input: input)
        }
        catch {
            print("ERROR: ", dump(error, name: "Deleting a bucket"))
            throw error
        }
    }

    /// Upload a file from local storage to the bucket.
    /// - Parameters:
    ///   - bucket: Name of the bucket to upload the file to.
    ///   - key: Name of the file to create.
    ///   - file: Path name of the file to upload.
    public func uploadFile(bucket: String, key: String, file: String) async
    throws {
        let fileUrl = URL(fileURLWithPath: file)
        do {
            let fileData = try Data(contentsOf: fileUrl)
            let dataStream = ByteStream.data(fileData)

            let input = PutObjectInput(
                body: dataStream,
                bucket: bucket,
                key: key
            )

            _ = try await client.putObject(input: input)
        }
        catch {
            print("ERROR: ", dump(error, name: "Putting an object. "))
            throw error
        }
    }
}
```

```
    }  
  }  
  
  /// Create a file in the specified bucket with the given name. The new  
  /// file's contents are uploaded from a `Data` object.  
  ///  
  /// - Parameters:  
  ///   - bucket: Name of the bucket to create a file in.  
  ///   - key: Name of the file to create.  
  ///   - data: A `Data` object to write into the new file.  
  public func createFile(bucket: String, key: String, withData data: Data)  
async throws {  
    let dataStream = ByteStream.data(data)  
  
    let input = PutObjectInput(  
      body: dataStream,  
      bucket: bucket,  
      key: key  
    )  
  
    do {  
      _ = try await client.putObject(input: input)  
    }  
    catch {  
      print("ERROR: ", dump(error, name: "Putting an object."))  
      throw error  
    }  
  }  
  
  /// Download the named file to the given directory on the local device.  
  ///  
  /// - Parameters:  
  ///   - bucket: Name of the bucket that contains the file to be copied.  
  ///   - key: The name of the file to copy from the bucket.  
  ///   - to: The path of the directory on the local device where you want to  
  ///     download the file.  
  public func downloadFile(bucket: String, key: String, to: String) async  
throws {  
    let fileUrl = URL(fileURLWithPath: to).appendingPathComponent(key)  
  
    let input = GetObjectInput(  
      bucket: bucket,  
      key: key  
    )  
  }
```



```
do {
    let output = try await client.getObject(input: input)

    guard let body = output.body else {
        throw HandlerError.getObjectBody("GetObjectInput missing body.")
    }

    guard let data = try await body.readData() else {
        throw HandlerError.readGetObjectBody("GetObjectInput unable to
read data.")
    }

    try data.write(to: fileUrl)
}
catch {
    print("ERROR: ", dump(error, name: "Downloading a file. "))
    throw error
}
}

/// Read the specified file from the given S3 bucket into a Swift
/// `Data` object.
///
/// - Parameters:
///   - bucket: Name of the bucket containing the file to read.
///   - key: Name of the file within the bucket to read.
///
/// - Returns: A `Data` object containing the complete file data.
public func readFile(bucket: String, key: String) async throws -> Data {
    let input = GetObjectInput(
        bucket: bucket,
        key: key
    )
    do {
        let output = try await client.getObject(input: input)

        guard let body = output.body else {
            throw HandlerError.getObjectBody("GetObjectInput missing body.")
        }

        guard let data = try await body.readData() else {
            throw HandlerError.readGetObjectBody("GetObjectInput unable to
read data.")
        }
    }
}
```

```
        return data
    }
    catch {
        print("ERROR: ", dump(error, name: "Reading a file. "))
        throw error
    }
}

/// Copy a file from one bucket to another.
///
/// - Parameters:
///   - sourceBucket: Name of the bucket containing the source file.
///   - name: Name of the source file.
///   - destBucket: Name of the bucket to copy the file into.
public func copyFile(from sourceBucket: String, name: String, to destBucket:
String) async throws {
    let srcUrl = ("\"(sourceBucket)/
\"(name)").addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)

    let input = CopyObjectInput(
        bucket: destBucket,
        copySource: srcUrl,
        key: name
    )
    do {
        _ = try await client.copyObject(input: input)
    }
    catch {
        print("ERROR: ", dump(error, name: "Copying an object. "))
        throw error
    }
}

/// Deletes the specified file from Amazon S3.
///
/// - Parameters:
///   - bucket: Name of the bucket containing the file to delete.
///   - key: Name of the file to delete.
///
public func deleteFile(bucket: String, key: String) async throws {
    let input = DeleteObjectInput(
        bucket: bucket,
```

```
        key: key
    )

    do {
        _ = try await client.deleteObject(input: input)
    }
    catch {
        print("ERROR: ", dump(error, name: "Deleting a file. "))
        throw error
    }
}

/// Returns an array of strings, each naming one file in the
/// specified bucket.
///
/// - Parameter bucket: Name of the bucket to get a file listing for.
/// - Returns: An array of `String` objects, each giving the name of
///           one file contained in the bucket.
public func listBucketFiles(bucket: String) async throws -> [String] {
    do {
        let input = ListObjectsV2Input(
            bucket: bucket
        )

        // Use "Paginated" to get all the objects.
        // This lets the SDK handle the 'continuationToken' in
        "ListObjectsV2Output".
        let output = client.listObjectsV2Paginated(input: input)
        var names: [String] = []

        for try await page in output {
            guard let objList = page.contents else {
                print("ERROR: listObjectsV2Paginated returned nil contents.")
                continue
            }

            for obj in objList {
                if let objName = obj.key {
                    names.append(objName)
                }
            }
        }
    }
}
```

```
        return names
    }
    catch {
        print("ERROR: ", dump(error, name: "Listing objects. "))
        throw error
    }
}
}
```

```
import AWSS3

import Foundation
import ServiceHandler
import ArgumentParser

/// The command-line arguments and options available for this
/// example command.
struct ExampleCommand: ParsableCommand {
    @Argument(help: "Name of the S3 bucket to create")
    var bucketName: String

    @Argument(help: "Pathname of the file to upload to the S3 bucket")
    var uploadSource: String

    @Argument(help: "The name (key) to give the file in the S3 bucket")
    var objName: String

    @Argument(help: "S3 bucket to copy the object to")
    var destBucket: String

    @Argument(help: "Directory where you want to download the file from the S3
bucket")
    var downloadDir: String

    static var configuration = CommandConfiguration(
        commandName: "s3-basics",
        abstract: "Demonstrates a series of basic AWS S3 functions.",
        discussion: ""
        Performs the following Amazon S3 commands:

        * `CreateBucket`
        * `PutObject`
    )
}
```

```
    * `GetObject`
    * `CopyObject`
    * `ListObjects`
    * `DeleteObjects`
    * `DeleteBucket`
    ""
)

/// Called by ``main()`` to do the actual running of the AWS
/// example.
func runAsync() async throws {
    let serviceHandler = try await ServiceHandler()

    // 1. Create the bucket.
    print("Creating the bucket \(bucketName)...")
    try await serviceHandler.createBucket(name: bucketName)

    // 2. Upload a file to the bucket.
    print("Uploading the file \(uploadSource)...")
    try await serviceHandler.uploadFile(bucket: bucketName, key: objName,
file: uploadSource)

    // 3. Download the file.
    print("Downloading the file \(objName) to \(downloadDir)...")
    try await serviceHandler.downloadFile(bucket: bucketName, key: objName,
to: downloadDir)

    // 4. Copy the file to another bucket.
    print("Copying the file to the bucket \(destBucket)...")
    try await serviceHandler.copyFile(from: bucketName, name: objName, to:
destBucket)

    // 5. List the contents of the bucket.

    print("Getting a list of the files in the bucket \(bucketName)")
    let fileList = try await serviceHandler.listBucketFiles(bucket:
bucketName)
    let numFiles = fileList.count
    if numFiles != 0 {
        print("\(numFiles) file\((numFiles > 1) ? "s" : "") in bucket
\(bucketName):")
        for name in fileList {
            print(" \(name)")
        }
    }
}
```

```
    } else {
        print("No files found in bucket \(bucketName)")
    }

    // 6. Delete the objects from the bucket.

    print("Deleting the file \(objName) from the bucket \(bucketName)...")
    try await serviceHandler.deleteFile(bucket: bucketName, key: objName)
    print("Deleting the file \(objName) from the bucket \(destBucket)...")
    try await serviceHandler.deleteFile(bucket: destBucket, key: objName)

    // 7. Delete the bucket.
    print("Deleting the bucket \(bucketName)...")
    try await serviceHandler.deleteBucket(name: bucketName)

    print("Done.")
}
}

//
// Main program entry point.
//
@main
struct Main {
    static func main() async {
        let args = Array(CommandLine.arguments.dropFirst())

        do {
            let command = try ExampleCommand.parse(args)
            try await command.runAsync()
        } catch {
            ExampleCommand.exit(withError: error)
        }
    }
}
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK als API-Swift-Referenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)

- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Aktionen für Amazon S3 mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie einzelne Amazon S3 S3-Aktionen mit AWS SDKs ausführen. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Diese Auszüge nennen Amazon S3 API und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Aktionen im Kontext finden Sie unter [Szenarien für die Verwendung von Amazon S3 AWS SDKs](#)

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der [Amazon Simple Storage Service API Reference](#).

Beispiele

- [Verwenden Sie AbortMultipartUpload mit einem AWS SDKoder CLI](#)
- [Verwenden Sie AbortMultipartUploads mit einem AWS SDKoder CLI](#)
- [Verwenden Sie CompleteMultipartUpload mit einem AWS SDKoder CLI](#)
- [Verwenden Sie CopyObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie CreateBucket mit einem AWS SDKoder CLI](#)
- [Verwenden Sie CreateMultiRegionAccessPoint mit einem AWS SDKoder CLI](#)
- [Verwenden Sie CreateMultipartUpload mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucket mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketAnalyticsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketCors mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketEncryption mit einem AWS SDKoder CLI](#)

- [Verwenden Sie DeleteBucketInventoryConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketLifecycle mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketMetricsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteBucketWebsite mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteObjectTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeleteObjects mit einem AWS SDKoder CLI](#)
- [Verwenden Sie DeletePublicAccessBlock mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAccelerateConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketAnalyticsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketCors mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketEncryption mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketInventoryConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLifecycleConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLocation mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketLogging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketMetricsConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketNotification mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketPolicyStatus mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketRequestPayment mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketVersioning mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetBucketWebsite mit einem AWS SDKoder CLI](#)

- [Verwenden Sie GetObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectAttributes mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectLegalHold mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectLockConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectRetention mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetObjectTagging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie GetPublicAccessBlock mit einem AWS SDKoder CLI](#)
- [Verwenden Sie HeadBucket mit einem AWS SDKoder CLI](#)
- [Verwenden Sie HeadObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBucketAnalyticsConfigurations mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBucketInventoryConfigurations mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListBuckets mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListMultipartUploads mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjectVersions mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjects mit einem AWS SDKoder CLI](#)
- [Verwenden Sie ListObjectsV2 mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketAccelerateConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketCors mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketEncryption mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketLifecycleConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketLogging mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketNotification mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketNotificationConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketPolicy mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketReplication mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketRequestPayment mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketTagging mit einem AWS SDKoder CLI](#)

- [Verwenden Sie PutBucketVersioning mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutBucketWebsite mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectAcl mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectLegalHold mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectLockConfiguration mit einem AWS SDKoder CLI](#)
- [Verwenden Sie PutObjectRetention mit einem AWS SDKoder CLI](#)
- [Verwenden Sie RestoreObject mit einem AWS SDKoder CLI](#)
- [Verwenden Sie SelectObjectContent mit einem AWS SDKoder CLI](#)
- [Verwenden Sie UploadPart mit einem AWS SDKoder CLI](#)

Verwenden Sie **AbortMultipartUpload** mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen die Verwendung `AbortMultipartUpload`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Löschen Sie unvollständige mehrteilige Uploads](#)
- [Arbeiten Sie mit Amazon S3 S3-Objektintegrität](#)

C++

SDK für C++

Note

Es gibt noch mehr dazu [GitHub](#). Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#!/ Abort a multipart upload to an S3 bucket.  
/*!  
    \param bucket: The name of the S3 bucket where the object will be uploaded.
```

```
\param key: The unique identifier (key) for the object within the S3 bucket.
\param uploadID: An upload ID string.
\param client: The S3 client instance used to perform the upload operation.
\return bool: Function succeeded.
*/

bool AwsDoc::S3::abortMultipartUpload(const Aws::String &bucket,
                                     const Aws::String &key,
                                     const Aws::String &uploadID,
                                     const Aws::S3::S3Client &client) {
    Aws::S3::Model::AbortMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);

    Aws::S3::Model::AbortMultipartUploadOutcome outcome =
        client.AbortMultipartUpload(request);

    if (outcome.IsSuccess()) {
        std::cout << "Multipart upload aborted." << std::endl;
    } else {
        std::cerr << "Error aborting multipart upload: " <<
outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API Einheiten finden Sie [AbortMultipartUpload](#) unter AWS SDK for C++ API Referenz.

CLI

AWS CLI

Um den angegebenen mehrteiligen Upload abubrechen

Mit dem folgenden `abort-multipart-upload` Befehl wird ein mehrteiliger Upload für den Schlüssel `multipart/01` im Bucket `my-bucket` abgebrochen.

```
aws s3api abort-multipart-upload \
  --bucket my-bucket \
  --key multipart/01 \
```

--upload-

id *dfRtDYU0WwCCcH43C3WfbkR0NycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P*

Die für diesen Befehl benötigte Upload-ID wird von ausgegeben `create-multipart-upload` und kann auch mit `list-multipart-uploads` abgerufen werden.

- API-Einheiten finden Sie [AbortMultipartUpload](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Mit diesem Befehl werden mehrteilige Uploads abgebrochen, die vor mehr als 5 Tagen erstellt wurden.

```
Remove-S3MultipartUpload -BucketName test-files -DaysBefore 5
```

Beispiel 2: Mit diesem Befehl werden mehrteilige Uploads abgebrochen, die vor dem 2. Januar 2014 erstellt wurden.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "Thursday, January 02, 2014"
```

Beispiel 3: Mit diesem Befehl werden mehrteilige Uploads abgebrochen, die vor dem 2. Januar 2014, 10:45:37 erstellt wurden.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "2014/01/02 10:45:37"
```

- [AbortMultipartUpload](#) Einzelheiten finden Sie unter API AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **AbortMultipartUploads** mit einem AWS SDK oder CLI

Das folgende Codebeispiel zeigt die Verwendung `AbortMultipartUploads`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Transfer;

/// <summary>
/// This example shows how to use the Amazon Simple Storage Service
/// (Amazon S3) to stop a multi-part upload process using the Amazon S3
/// TransferUtility.
/// </summary>
public class AbortMPU
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the S3 client object's constructor.
        // For example: RegionEndpoint.USWest2.
        IAmazonS3 client = new AmazonS3Client();

        await AbortMPUAsync(client, bucketName);
    }

    /// <summary>
    /// Cancels the multi-part copy process.
    /// </summary>
    /// <param name="client">The initialized client object used to create
    /// the TransferUtility object.</param>
    /// <param name="bucketName">The name of the S3 bucket where the
    /// multi-part copy operation is in progress.</param>
}
```

```
public static async Task AbortMPUAsync(IAmazonS3 client, string
bucketName)
{
    try
    {
        var transferUtility = new TransferUtility(client);

        // Cancel all in-progress uploads initiated before the specified
date.
        await transferUtility.AbortMultipartUploadsAsync(
            bucketName, DateTime.Now.AddDays(-7));
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine($"Error: {e.Message}");
    }
}
}
```

- API-Einheiten finden Sie [AbortMultipartUploads](#) unter AWS SDK for .NET API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **CompleteMultipartUpload** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `CompleteMultipartUpload`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erstellen einer mehrteiligen Kopie](#)
- [Durchführen eines mehrteiligen Uploads](#)
- [Verwenden der Prüfsummen](#)
- [Arbeiten Sie mit Amazon S3 S3-Objektintegrität](#)

C++

SDK für C++

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
///  
/*!  
    \param bucket: The name of the S3 bucket where the object will be uploaded.  
    \param key: The unique identifier (key) for the object within the S3 bucket.  
    \param uploadID: An upload ID string.  
    \param parts: A vector of CompleteParts.  
    \param client: The S3 client instance used to perform the upload operation.  
    \return CompleteMultipartUploadOutcome: The request outcome.  
*/  
Aws::S3::Model::CompleteMultipartUploadOutcome  
    AwsDoc::S3::completeMultipartUpload(const Aws::String &bucket,  
  
    const Aws::String &key,  
  
    const Aws::String &uploadID,  
  
    const Aws::Vector<Aws::S3::Model::CompletedPart> &parts,  
  
    const Aws::S3::S3Client &client) {  
    Aws::S3::Model::CompletedMultipartUpload completedMultipartUpload;  
    completedMultipartUpload.SetParts(parts);  
  
    Aws::S3::Model::CompleteMultipartUploadRequest request;  
    request.SetBucket(bucket);  
    request.SetKey(key);  
    request.SetUploadId(uploadID);  
    request.SetMultipartUpload(completedMultipartUpload);  
  
    Aws::S3::Model::CompleteMultipartUploadOutcome outcome =  
        client.CompleteMultipartUpload(request);  
  
    if (!outcome.IsSuccess()) {
```

```

        std::cerr << "Error completing multipart upload: " <<
outcome.GetError().GetMessage() << std::endl;
    }
    return outcome;
}

```

- API-Einheiten finden Sie [CompleteMultipartUpload](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Mit dem folgenden Befehl wird ein mehrteiliger Upload für den Schlüssel `multipart/01` im Bucket `my-bucket` abgeschlossen:

```

aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket my-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P

```

Die für diesen Befehl benötigte Upload-ID wird von `create-multipart-upload` ausgegeben und kann auch mit `list-multipart-uploads` abgerufen werden.

Die mehrteilige Upload-Option im obigen Befehl hat eine JSON-Struktur, die die Teile des mehrteiligen Uploads beschreibt, die wieder zur vollständigen Datei zusammengesetzt werden sollen. In diesem Beispiel wird das `file://`-Präfix verwendet, um die JSON-Struktur aus einer Datei im lokalen Ordner mit dem Namen `mpustruct`

`mpustruct`:

```

{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",

```



```
    "PartNumber": 3
  }
]
}
```

Der ETag Wert für jeden Teil, der hochgeladen wird, wird jedes Mal ausgegeben, wenn Sie ein Teil mit dem `upload-part` Befehl hochladen. Er kann auch durch Aufrufen abgerufen `list-parts` oder berechnet werden, indem die MD5 Prüfsumme jedes Teils genommen wird.

Ausgabe:

```
{
  "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
  "Bucket": "my-bucket",
  "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
  "Key": "multipart/01"
}
```

- API Einzelheiten finden Sie unter [CompleteMultipartUpload](#) AWS CLI Befehlsreferenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// upload_parts: Vec<aws_sdk_s3::types::CompletedPart>
let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
    .set_parts(Some(upload_parts))
    .build();

let _complete_multipart_upload_res = client
    .complete_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .multipart_upload(completed_multipart_upload)
```

```
.upload_id(upload_id)
.send()
.await?;
```

```
// Create a multipart upload. Use UploadPart and CompleteMultipartUpload to
// upload the file.
let multipart_upload_res: CreateMultipartUploadOutput = client
    .create_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .send()
    .await?;

let upload_id = multipart_upload_res.upload_id().ok_or(S3ExampleError::new(
    "Missing upload_id after CreateMultipartUpload",
))?;
```

```
let mut upload_parts: Vec<aws_sdk_s3::types::CompletedPart> = Vec::new();

for chunk_index in 0..chunk_count {
    let this_chunk = if chunk_count - 1 == chunk_index {
        size_of_last_chunk
    } else {
        CHUNK_SIZE
    };
    let stream = ByteStream::read_from()
        .path(path)
        .offset(chunk_index * CHUNK_SIZE)
        .length(Length::Exact(this_chunk))
        .build()
        .await
        .unwrap();

    // Chunk index needs to start at 0, but part numbers start at 1.
    let part_number = (chunk_index as i32) + 1;
    let upload_part_res = client
        .upload_part()
        .key(&key)
        .bucket(&bucket_name)
        .upload_id(upload_id)
        .body(stream)
```

```
        .part_number(part_number)
        .send()
        .await?;

upload_parts.push(
    CompletedPart::builder()
        .e_tag(upload_part_res.e_tag.unwrap_or_default())
        .part_number(part_number)
        .build(),
);
}
```

- API-Einheiten finden Sie [CompleteMultipartUpload](#) unter AWS SDK als API-Referenz auf Rust.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **CopyObject** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `CopyObject`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erlernen Sie die Grundlagen](#)
- [Erste Schritte mit der Verschlüsselung](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

public class CopyObject
{
    public static async Task Main()
    {
        // Specify the AWS Region where your buckets are located if it is
        // different from the AWS Region of the default user.
        IAmazonS3 s3Client = new AmazonS3Client();

        // Remember to change these values to refer to your Amazon S3
objects.
        string sourceBucketName = "amzn-s3-demo-bucket1";
        string destinationBucketName = "amzn-s3-demo-bucket2";
        string sourceObjectKey = "testfile.txt";
        string destinationObjectKey = "testfilecopy.txt";

        Console.WriteLine($"Copying {sourceObjectKey} from {sourceBucketName}
to ");
        Console.WriteLine($"{destinationBucketName} as
{destinationObjectKey}");

        var response = await CopyingObjectAsync(
            s3Client,
            sourceObjectKey,
            destinationObjectKey,
            sourceBucketName,
            destinationBucketName);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine("\nCopy complete.");
        }
    }

    /// <summary>
    /// This method calls the AWS SDK for .NET to copy an
    /// object from one Amazon S3 bucket to another.
    /// </summary>
    /// <param name="client">The Amazon S3 client object.</param>
}
```

```
    /// <param name="sourceKey">The name of the object to be copied.</param>
    /// <param name="destinationKey">The name under which to save the copy.</
param>
    /// <param name="sourceBucketName">The name of the Amazon S3 bucket
    /// where the file is located now.</param>
    /// <param name="destinationBucketName">The name of the Amazon S3
    /// bucket where the copy should be saved.</param>
    /// <returns>Returns a CopyObjectResponse object with the results from
    /// the async call.</returns>
    public static async Task<CopyObjectResponse> CopyingObjectAsync(
        IAmazonS3 client,
        string sourceKey,
        string destinationKey,
        string sourceBucketName,
        string destinationBucketName)
    {
        var response = new CopyObjectResponse();
        try
        {
            var request = new CopyObjectRequest
            {
                SourceBucket = sourceBucketName,
                SourceKey = sourceKey,
                DestinationBucket = destinationBucketName,
                DestinationKey = destinationKey,
            };
            response = await client.CopyObjectAsync(request);
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error copying object: '{ex.Message}'");
        }

        return response;
    }
}
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response

    response=$(aws s3api copy-object \
        --bucket "$bucket_name" \
        --copy-source "$bucket_name/$source_key" \
```

```

--key "$destination_key")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
    return 1
fi
}

```

- API-Einheiten finden Sie [CopyObject](#) unter AWS CLI Befehlsreferenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

bool AwsDoc::S3::copyObject(const Aws::String &objectKey, const Aws::String
&fromBucket, const Aws::String &toBucket,
                          const Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::CopyObjectRequest request;

    request.WithCopySource(fromBucket + "/" + objectKey)
           .WithKey(objectKey)
           .WithBucket(toBucket);

    Aws::S3::Model::CopyObjectOutcome outcome = client.CopyObject(request);
    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: copyObject: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
        std::endl;
    } else {
        std::cout << "Successfully copied " << objectKey << " from " <<
        fromBucket <<

```

```
        " to " << toBucket << "." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API Einzelheiten finden Sie [CopyObject](#) unter AWS SDK for C++ API Referenz.

CLI

AWS CLI

Der folgende Befehl kopiert ein Objekt von bucket-1 nach bucket-2:

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2
```

Ausgabe:

```
{
  "CopyObjectResult": {
    "LastModified": "2015-11-10T01:07:25.000Z",
    "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
  },
  "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifM1kA"
}
```

- API Einzelheiten finden Sie [CopyObject](#) unter AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).


```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// CopyToBucket copies an object in a bucket to another bucket.
func (basics BucketBasics) CopyToBucket(sourceBucket string, destinationBucket
string, objectKey string) error {
    _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
        Bucket:      aws.String(destinationBucket),
        CopySource:  aws.String(fmt.Sprintf("%v/%v", sourceBucket, objectKey)),
        Key:         aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't copy object from %v:%v to %v:%v. Here's why: %v\n",
            sourceBucket, objectKey, destinationBucket, objectKey, err)
    }
    return err
}
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Kopieren eines Objekts mit einem [S3Client](#).

```
/**
 * Asynchronously copies an object from one S3 bucket to another.
 *
 * @param fromBucket the name of the source S3 bucket
 * @param objectKey the key (name) of the object to be copied
 * @param toBucket the name of the destination S3 bucket
 * @return a {@link CompletableFuture} that completes with the copy result as
 * a {@link String}
 * @throws RuntimeException if the URL could not be encoded or an S3
 * exception occurred during the copy
 */
public CompletableFuture<String> copyBucketObjectAsync(String fromBucket,
String objectKey, String toBucket) {
    CopyObjectRequest copyReq = CopyObjectRequest.builder()
        .sourceBucket(fromBucket)
        .sourceKey(objectKey)
        .destinationBucket(toBucket)
        .destinationKey(objectKey)
        .build();

    CompletableFuture<CopyObjectResponse> response =
getAsyncClient().copyObject(copyReq);
    response.whenComplete((copyRes, ex) -> {
        if (copyRes != null) {
            logger.info("The " + objectKey + " was copied to " + toBucket);
        } else {
            throw new RuntimeException("An S3 exception occurred during
copy", ex);
        }
    });

    return response.thenApply(CopyObjectResponse::copyObjectResult)
        .thenApply(Object::toString);
}
```

Verwenden Sie einen [S3 TransferManager](#), um [ein Objekt von einem Bucket in einen anderen zu kopieren](#). Sehen Sie sich die [vollständige Datei](#) an und [testen](#) Sie sie.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

```
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.CopyObjectRequest;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedCopy;
import software.amazon.awssdk.transfer.s3.model.Copy;
import software.amazon.awssdk.transfer.s3.model.CopyRequest;

import java.util.UUID;

    public String copyObject(S3TransferManager transferManager, String
    bucketName,
        String key, String destinationBucket, String destinationKey) {
        CopyObjectRequest copyObjectRequest = CopyObjectRequest.builder()
            .sourceBucket(bucketName)
            .sourceKey(key)
            .destinationBucket(destinationBucket)
            .destinationKey(destinationKey)
            .build();

        CopyRequest copyRequest = CopyRequest.builder()
            .copyObjectRequest(copyObjectRequest)
            .build();

        Copy copy = transferManager.copy(copyRequest);

        CompletedCopy completedCopy = copy.completionFuture().join();
        return completedCopy.response().copyObjectResult().eTag();
    }
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Kopieren Sie das Objekt.

```
import { S3Client, CopyObjectCommand } from "@aws-sdk/client-s3";

const client = new S3Client({});

/**
 * Copy an Amazon S3 object from one bucket to another.
 */
export const main = async () => {
  const command = new CopyObjectCommand({
    CopySource: "SOURCE_BUCKET/SOURCE_OBJECT_KEY",
    Bucket: "DESTINATION_BUCKET",
    Key: "NEW_OBJECT_KEY",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun copyBucketObject(
  fromBucket: String,
  objectKey: String,
  toBucket: String,
```

```
) {
    var encodedUrl = ""
    try {
        encodedUrl = URLEncoder.encode("$fromBucket/$objectKey",
StandardCharsets.UTF_8.toString())
    } catch (e: UnsupportedEncodingException) {
        println("URL could not be encoded: " + e.message)
    }

    val request =
        CopyObjectRequest {
            copySource = encodedUrl
            bucket = toBucket
            key = objectKey
        }
    S3Client { region = "us-east-1" }.use { s3 ->
        s3.copyObject(request)
    }
}
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK als API Kotlin-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Einfache Kopie eines Objekts.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $folder = "copied-folder";
    $this->s3client->copyObject([
        'Bucket' => $this->bucketName,
        'CopySource' => "$this->bucketName/$fileName",
```

```
        'Key' => "$folder/$fileName-copy",
    ]);
    echo "Copied $fileName to $folder/$fileName-copy.\n";
} catch (Exception $exception) {
    echo "Failed to copy $fileName with error: " . $exception-
>getMessage();
    exit("Please fix error with object copying before continuing.");
}
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl kopiert das Objekt "sample.txt" aus dem Bucket „test-files“ in denselben Bucket, jedoch mit dem neuen Schlüssel "sample-copy.txt“.

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-copy.txt
```

Beispiel 2: Dieser Befehl kopiert das Objekt "sample.txt" aus dem Bucket „test-files“ in den Bucket „backup-files“ mit dem Schlüssel "sample-copy.txt“.

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-copy.txt -DestinationBucket backup-files
```

Beispiel 3: Dieser Befehl lädt das Objekt "sample.txt" aus dem Bucket „test-files“ in eine lokale Datei mit dem Namen "local-sample.txt" herunter.

```
Copy-S3Object -BucketName test-files -Key sample.txt -LocalFile local-sample.txt
```

Beispiel 4: Lädt das einzelne Objekt in die angegebene Datei herunter. Die heruntergeladene Datei befindet sich unter c:\downloads\data\archive.zip

```
Copy-S3Object -BucketName test-files -Key data/archive.zip -LocalFolder c:\downloads
```

Beispiel 5: Lädt alle Objekte, die dem angegebenen key prefix entsprechen, in den lokalen Ordner herunter. Die relative Schlüsselhierarchie wird als Unterordner im gesamten Download-Speicherort beibehalten.

```
Copy-S3Object -BucketName test-files -KeyPrefix data -LocalFolder c:\downloads
```

- API-Einheiten finden Sie unter [CopyObject](#) AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
                               that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def copy(self, dest_object):
        """
        Copies the object to another bucket.

        :param dest_object: The destination object initialized with a bucket and
        key.
                               This is a Boto3 Object resource.
        """
        try:
            dest_object.copy_from(
```

```
        CopySource={"Bucket": self.object.bucket_name, "Key":
self.object.key}
    )
    dest_object.wait_until_exists()
    logger.info(
        "Copied object from %s:%s to %s:%s.",
        self.object.bucket_name,
        self.object.key,
        dest_object.bucket_name,
        dest_object.key,
    )
except ClientError:
    logger.exception(
        "Couldn't copy object from %s/%s to %s/%s.",
        self.object.bucket_name,
        self.object.key,
        dest_object.bucket_name,
        dest_object.key,
    )
    raise
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Kopieren Sie ein Objekt.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  attr_reader :source_object
```



```
# @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
used as the source object for
#
#                                     copy actions.
def initialize(source_object)
  @source_object = source_object
end

# Copy the source object to the specified target bucket and rename it with the
target key.
#
# @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
object is copied.
# @param target_object_key [String] The key to give the copy of the object.
# @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
nil.
def copy_object(target_bucket, target_object_key)
  @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
  target_bucket.object(target_object_key)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's
why: #{e.message}"
end
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Kopieren Sie ein Objekt und fügen Sie dem Zielobjekt eine serverseitige Verschlüsselung hinzu.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
  # used as the source object for
  #
  #
  # @return [Aws::S3::Object] The copied object when successful; otherwise,
  # nil.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the
  # target key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
  # object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
  # nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's
why: #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"
end
```

```
source_bucket = Aws::S3::Bucket.new(source_bucket_name)
wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
target_bucket = Aws::S3::Bucket.new(target_bucket_name)
target_object = wrapper.copy_object(target_bucket, target_key,
target_encryption)
return unless target_object

puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key} and "\
    "encrypted the target with #{target_object.server_side_encryption}
encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// Copy an object from one bucket to another.
pub async fn copy_object(
    client: &aws_sdk_s3::Client,
    source_bucket: &str,
    destination_bucket: &str,
    source_object: &str,
    destination_object: &str,
) -> Result<(), S3ExampleError> {
    let source_key = format!("{source_bucket}/{source_object}");
    let response = client
        .copy_object()
        .copy_source(&source_key)
        .bucket(destination_bucket)
```

```

        .key(destination_object)
        .send()
        .await?;

println!(
    "Copied from {source_key} to {destination_bucket}/{destination_object}
with etag {}",
    response
        .copy_object_result
        .unwrap_or_else(||
aws_sdk_s3::types::CopyObjectResult::builder().build())
        .e_tag()
        .unwrap_or("missing")
    );
    Ok(())
}

```

- API Einzelheiten finden Sie [CopyObject](#) unter AWS SDK als API Referenz auf Rust.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

TRY.
  lo_s3->copyobject(
    iv_bucket = iv_dest_bucket
    iv_key = iv_dest_object
    iv_copysource = |{ iv_src_bucket }/{ iv_src_object }|
  ).
  MESSAGE 'Object copied to another bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK als SAP ABAP API Referenz.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3

public func copyFile(from sourceBucket: String, name: String, to destBucket:
String) async throws {
    let srcUrl = ("\"(sourceBucket)/
\"(name)\"").addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)

    let input = CopyObjectInput(
        bucket: destBucket,
        copySource: srcUrl,
        key: name
    )
    do {
        _ = try await client.copyObject(input: input)
    }
    catch {
        print("ERROR: ", dump(error, name: "Copying an object. "))
        throw error
    }
}
```

- API-Einheiten finden Sie [CopyObject](#) unter AWS SDK als API Swift-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **CreateBucket** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `CreateBucket`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erlernen Sie die Grundlagen](#)
- [Mit versionierten Objekten arbeiten](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Shows how to create a new Amazon S3 bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket to create.</param>
/// <returns>A boolean value representing the success or failure of
/// the bucket creation process.</returns>
public static async Task<bool> CreateBucketAsync(IAmazonS3 client, string
bucketName)
{
    try
    {
        var request = new PutBucketRequest
        {
            BucketName = bucketName,
```

```
        UseClientRegion = true,
    };

    var response = await client.PutBucketAsync(request);
    return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error creating bucket: '{ex.Message}'");
    return false;
}
}
```

Erstellen Sie einen Bucket mit aktivierter Objektsperre.

```
/// <summary>
/// Create a new Amazon S3 bucket with object lock actions.
/// </summary>
/// <param name="bucketName">The name of the bucket to create.</param>
/// <param name="enableObjectLock">True to enable object lock on the
bucket.</param>
/// <returns>True if successful.</returns>
public async Task<bool> CreateBucketWithObjectLock(string bucketName, bool
enableObjectLock)
{
    Console.WriteLine($"\\tCreating bucket {bucketName} with object lock
{enableObjectLock}.");
    try
    {
        var request = new PutBucketRequest
        {
            BucketName = bucketName,
            UseClientRegion = true,
            ObjectLockEnabledForBucket = enableObjectLock,
        };

        var response = await _amazonS3.PutBucketAsync(request);

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
```

```

    {
        Console.WriteLine($"Error creating bucket: '{ex.Message}'");
        return false;
    }
}

```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function create-bucket

```



```
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#     -b bucket_name  -- The name of the bucket to create.
#     -r region_code  -- The code for an AWS Region in which to
#                       create the bucket.
#
# Returns:
#     The URL of the bucket that was created.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function create_bucket"
        echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
        echo "  -b bucket_name    The name of the bucket. It must be globally
unique."
        echo "  [-r region_code]   The code for an AWS Region in which the bucket is
created."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "b:r:h" option; do
        case "${option}" in
            b) bucket_name="${OPTARG}" ;;
            r) region_code="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}
```

```
    esac
done

if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
fi

local bucket_config_arg
# A location constraint for "us-east-1" returns an error.
if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
iecho "    Bucket name:  $bucket_name"
iecho "    Region code:  $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
fi


# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
}
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS CLI Befehlsreferenz.

C++

SDK für C++

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::createBucket(const Aws::String &bucketName,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::CreateBucketRequest request;
    request.SetBucket(bucketName);

    if (clientConfig.region != "us-east-1") {
        Aws::S3::Model::CreateBucketConfiguration createBucketConfig;
        createBucketConfig.SetLocationConstraint(
            Aws::S3::Model::BucketLocationConstraintMapper::GetBucketLocationConstraintForName(
                clientConfig.region));
        request.SetCreateBucketConfiguration(createBucketConfig);
    }

    Aws::S3::Model::CreateBucketOutcome outcome = client.CreateBucket(request);
    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error: createBucket: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
            std::endl;
    } else {
        std::cout << "Created bucket " << bucketName <<
            " in the specified AWS Region." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API Einzelheiten finden Sie [CreateBucket](#) unter AWS SDK for C++ API Referenz.

CLI

AWS CLI

Beispiel 1: Um einen Bucket zu erstellen

Im folgenden `create-bucket` Beispiel wird ein Bucket mit dem Namen `my-bucket`:

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1
```

Ausgabe:

```
{  
  "Location": "/my-bucket"  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Buckets](#) im Amazon S3 S3-Benutzerhandbuch.

Beispiel 2: Um einen Bucket zu erstellen, bei dem der Besitzer erzwungen wird

Im folgenden `create-bucket` Beispiel wird ein Bucket mit dem Namen `my-bucket` die Einstellung `Bucket Owner enforced` für S3 Object Ownership verwendet.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1 \  
  --object-ownership BucketOwnerEnforced
```

Ausgabe:

```
{  
  "Location": "/my-bucket"  
}
```

Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs](#) im Amazon S3 S3-Benutzerhandbuch.

Beispiel 3: Um einen Bucket außerhalb der Region `us-east-1` zu erstellen

Im folgenden `create-bucket` Beispiel wird ein Bucket mit dem Namen der Region erstellt. `my-bucket eu-west-1` Für Regionen außerhalb von `us-east-1` müssen die entsprechenden `LocationConstraint` Felder angegeben werden, um den Bucket in der gewünschten Region zu erstellen.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region eu-west-1 \  
  --create-bucket-configuration LocationConstraint=eu-west-1
```

Ausgabe:

```
{  
  "Location": "http://my-bucket.s3.amazonaws.com/"  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Buckets](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateBucket](#) unter AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie einen Bucket mit Standardkonfiguration.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)  
// actions  
// used in the examples.  
// It contains S3Client, an Amazon S3 service client that is used to perform  
// bucket  
// and object actions.
```

```
type BucketBasics struct {
    S3Client *s3.Client
}

// CreateBucket creates a bucket with the specified name in the specified Region.
func (basics BucketBasics) CreateBucket(name string, region string) error {
    _, err := basics.S3Client.CreateBucket(context.TODO(), &s3.CreateBucketInput{
        Bucket: aws.String(name),
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            LocationConstraint: types.BucketLocationConstraint(region),
        },
    })
    if err != nil {
        log.Printf("Couldn't create bucket %v in Region %v. Here's why: %v\n",
            name, region, err)
    }
    return err
}
```

Erstellen Sie einen Bucket mit Objektsperre und warten Sie, bis er existiert.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
    S3Client *s3.Client
    S3Manager *manager.Uploader
}

// CreateBucketWithLock creates a new S3 bucket with optional object locking
// enabled
// and waits for the bucket to exist before returning.
func (actor S3Actions) CreateBucketWithLock(ctx context.Context, bucket string,
    region string, enableObjectLock bool) (string, error) {
    input := &s3.CreateBucketInput{
        Bucket: aws.String(bucket),
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            LocationConstraint: types.BucketLocationConstraint(region),
        },
    }
```

```
    },
  }

  if enableObjectLock {
    input.ObjectLockEnabledForBucket = aws.Bool(true)
  }

  _, err := actor.S3Client.CreateBucket(ctx, input)
  if err != nil {
    var owned *types.BucketAlreadyOwnedByYou
    var exists *types.BucketAlreadyExists
    if errors.As(err, &owned) {
      log.Printf("You already own bucket %s.\n", bucket)
      err = owned
    } else if errors.As(err, &exists) {
      log.Printf("Bucket %s already exists.\n", bucket)
      err = exists
    }
  } else {
    err = s3.NewBucketExistsWaiter(actor.S3Client).Wait(
      ctx, &s3.HeadBucketInput{Bucket: aws.String(bucket)}, time.Minute)
    if err != nil {
      log.Printf("Failed attempt to wait for bucket %s to exist.\n", bucket)
    }
  }

  return bucket, err
}
```

- API Einzelheiten finden Sie [CreateBucket](#) unter AWS SDK for Go API Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie einen Bucket.

```
/**
 * Creates an S3 bucket asynchronously.
 *
 * @param bucketName the name of the S3 bucket to create
 * @return a {@link CompletableFuture} that completes when the bucket is
 * created and ready
 * @throws RuntimeException if there is a failure while creating the bucket
 */
public CompletableFuture<Void> createBucketAsync(String bucketName) {
    CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
        .bucket(bucketName)
        .build();

    CompletableFuture<CreateBucketResponse> response =
getAsyncClient().createBucket(bucketRequest);
    return response.thenCompose(resp -> {
        S3AsyncWaiter s3Waiter = getAsyncClient().waiter();
        HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
            .bucket(bucketName)
            .build();

        CompletableFuture<WaiterResponse<HeadBucketResponse>>
waiterResponseFuture =
            s3Waiter.waitUntilBucketExists(bucketRequestWait);
        return waiterResponseFuture.thenAccept(waiterResponse -> {
            waiterResponse.matched().response().ifPresent(headBucketResponse
-> {
                logger.info(bucketName + " is ready");
            });
        });
    }).whenComplete((resp, ex) -> {
        if (ex != null) {
            throw new RuntimeException("Failed to create bucket", ex);
        }
    });
}
```

Erstellen Sie einen Bucket mit aktivierter Objektsperre.

```
// Create a new Amazon S3 bucket with object lock options.
```



```
public void createBucketWithLockOptions(boolean enableObjectLock, String
bucketName) {
    S3Waiter s3Waiter = getClient().waiter();
    CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
        .bucket(bucketName)
        .objectLockEnabledForBucket(enableObjectLock)
        .build();

    getClient().createBucket(bucketRequest);
    HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
        .bucket(bucketName)
        .build();

    // Wait until the bucket is created and print out the response.
    s3Waiter.waitUntilBucketExists(bucketRequestWait);
    System.out.println(bucketName + " is ready");
}
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie den Bucket.

```
import { CreateBucketCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
    const command = new CreateBucketCommand({
        // The name of the bucket. Bucket names are unique and have several other
        constraints.
    });
```

```
// See https://docs.aws.amazon.com/AmazonS3/latest/userguide/
bucketnamingrules.html
    Bucket: "bucket-name",
  });

  try {
    const { Location } = await client.send(command);
    console.log(`Bucket created with location ${Location}`);
  } catch (err) {
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [CreateBucket](#) in AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun createNewBucket(bucketName: String) {
    val request =
        CreateBucketRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        println("$bucketName is ready")
    }
}
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK als API Kotlin-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie einen Bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $this->s3client->createBucket([
        'Bucket' => $this->bucketName,
        'CreateBucketConfiguration' => ['LocationConstraint' => $region],
    ]);
    echo "Created bucket named: $this->bucketName \n";
} catch (Exception $exception) {
    echo "Failed to create bucket $this->bucketName with error: " .
    $exception->getMessage();
    exit("Please fix error with bucket creation before continuing.");
}
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK for PHP API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie einen Bucket mit Standardeinstellungen.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                        that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def create(self, region_override=None):
        """
        Create an Amazon S3 bucket in the default Region for the account or in
        the
        specified Region.

        :param region_override: The Region in which to create the bucket. If this
        is
                                not specified, the Region configured in your
        shared
                                credentials is used.
        """
        if region_override is not None:
            region = region_override
        else:
            region = self.bucket.meta.client.meta.region_name
        try:
            self.bucket.create(CreateBucketConfiguration={"LocationConstraint":
            region})

            self.bucket.wait_until_exists()
            logger.info("Created bucket '%s' in region=%s", self.bucket.name,
            region)
        except ClientError as error:
            logger.exception(
                "Couldn't create bucket named '%s' in region=%s.",
                self.bucket.name,
                region,
            )
            raise error
```

Erstellen Sie einen versionierten Bucket mit einer Lebenszyklus-Konfiguration.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
    lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                "LocationConstraint": s3.meta.client.meta.region_name
            },
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
```

```
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                "Rules": [
                    {
                        "Status": "Enabled",
                        "Prefix": prefix,
                        "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration},
                    }
                ]
            }
        )
        logger.info(
            "Configured lifecycle to expire noncurrent versions after %s days "
            "on bucket %s.",
            expiration,
            bucket.name,
        )
    except ClientError as error:
        logger.warning(
            "Couldn't configure lifecycle on bucket %s because %s. "
            "Continuing anyway.",
            bucket.name,
            error,
        )

    return bucket
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK für Python (Boto3) API -Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name.
  # This is a client-side object until
  # create is called.
  def initialize(bucket)
    @bucket = bucket
  end

  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)
    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
    false
  end

  # Gets the Region where the bucket is located.
  #
  # @return [String] The location of the bucket.
  def location
    if @bucket.nil?
      "None. You must create a bucket before you can get its location!"
    else

```

```

    @bucket.client.get_bucket_location(bucket:
@bucket.name).location_constraint
    end
    rescue Aws::Errors::ServiceError => e
      "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
    end
  end
end

# Example usage:
def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("amzn-s3-demo-bucket-
#{@Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{@wrapper.bucket.name}."
  puts "Your bucket's region is: #{@wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__

```

- API Einzelheiten finden Sie [CreateBucket](#) unter AWS SDK for Ruby API Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

pub async fn create_bucket(
  client: &aws_sdk_s3::Client,
  bucket_name: &str,
  region: &aws_config::Region,
) -> Result<Option<aws_sdk_s3::operation::create_bucket::CreateBucketOutput>,
S3ExampleError> {
  let constraint =
aws_sdk_s3::types::BucketLocationConstraint::from(region.to_string().as_str());

```



```
let cfg = aws_sdk_s3::types::CreateBucketConfiguration::builder()
    .location_constraint(constraint)
    .build();
let create = client
    .create_bucket()
    .create_bucket_configuration(cfg)
    .bucket(bucket_name)
    .send()
    .await;

// BucketAlreadyExists and BucketAlreadyOwnedByYou are not problems for this
task.
create.map(Some).or_else(|err| {
    if err
        .as_service_error()
        .map(|se| se.is_bucket_already_exists() ||
se.is_bucket_already_owned_by_you())
        == Some(true)
    {
        Ok(None)
    } else {
        Err(S3ExampleError::from(err))
    }
}))
}
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK als API Referenz auf Rust.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

TRY.

```
lo_s3->createbucket(
    iv_bucket = iv_bucket_name
```

```
    ).
    MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK als SAP ABAP API Referenz.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3

public func createBucket(name: String) async throws {
    var input = CreateBucketInput(
        bucket: name
    )

    // For regions other than "us-east-1", you must set the
    locationConstraint in the createBucketConfiguration.
    // For more information, see LocationConstraint in the S3 API guide.
    // https://docs.aws.amazon.com/AmazonS3/latest/API/
API_CreateBucket.html#API_CreateBucket_RequestBody
    if let region = configuration.region {
        if region != "us-east-1" {
            input.createBucketConfiguration =
S3ClientTypes.CreateBucketConfiguration(locationConstraint:
S3ClientTypes.BucketLocationConstraint(rawValue: region))
        }
    }

    do {
```

```
        _ = try await client.createBucket(input: input)
    }
    catch let error as BucketAlreadyOwnedByYou {
        print("The bucket '\(name)' already exists and is owned by you. You
may wish to ignore this exception.")
        throw error
    }
    catch {
        print("ERROR: ", dump(error, name: "Creating a bucket"))
        throw error
    }
}
```

- API-Einheiten finden Sie [CreateBucket](#) unter AWS SDK als API-Swift-Referenz.


Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **CreateMultiRegionAccessPoint** mit einem AWS SDK oder CLI

Das folgende Codebeispiel zeigt die Verwendung `CreateMultiRegionAccessPoint`.

Kotlin

SDK für Kotlin

 Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Konfigurieren Sie den S3-Kontrollclient so, dass er eine Anfrage an die Region `us-west-2` sendet.

```
suspend fun createS3ControlClient(): S3ControlClient {
    // Configure your S3ControlClient to send requests to US West
(Oregon).
    val s3Control = S3ControlClient.fromEnvironment {
```

```

        region = "us-west-2"
    }
    return s3Control
}

```

Erstellen Sie den Access Point für mehrere Regionen.

```

suspend fun createMrap(
    s3Control: S3ControlClient,
    accountIdParam: String,
    bucketName1: String,
    bucketName2: String,
    mrapName: String,
): String {
    println("Creating MRAP ...")
    val createMrapResponse: CreateMultiRegionAccessPointResponse =
        s3Control.createMultiRegionAccessPoint {
            accountId = accountIdParam
            clientToken = UUID.randomUUID().toString()
            details {
                name = mrapName
                regions = listOf(
                    Region {
                        bucket = bucketName1
                    },
                    Region {
                        bucket = bucketName2
                    },
                )
            }
        }
    val requestToken: String? = createMrapResponse.requestTokenArn

    // Use the request token to check for the status of the
    CreateMultiRegionAccessPoint operation.
    if (requestToken != null) {
        waitForSucceededStatus(s3Control, requestToken, accountIdParam)
        println("MRAP created")
    }

    val getMrapResponse =
        s3Control.getMultiRegionAccessPoint(

```

```

        input = GetMultiRegionAccessPointRequest {
            accountId = accountIdParam
            name = mrapName
        },
    )
    val mrapAlias = getMrapResponse.accessPoint?.alias
    return "arn:aws:s3:::$accountIdParam:accesspoint/$mrapAlias"
}

```

Warten Sie, bis der Access Point für mehrere Regionen verfügbar ist.

```

suspend fun waitForSucceededStatus(
    s3Control: S3ControlClient,
    requestToken: String,
    accountIdParam: String,
    timeBetweenChecks: Duration = 1.minutes,
) {
    var describeResponse: DescribeMultiRegionAccessPointOperationResponse
    describeResponse = s3Control.describeMultiRegionAccessPointOperation(
        input = DescribeMultiRegionAccessPointOperationRequest {
            accountId = accountIdParam
            requestTokenArn = requestToken
        },
    )

    var status: String? = describeResponse.asyncOperation?.requestStatus
    while (status != "SUCCEEDED") {
        delay(timeBetweenChecks)
        describeResponse =
s3Control.describeMultiRegionAccessPointOperation(
            input = DescribeMultiRegionAccessPointOperationRequest {
                accountId = accountIdParam
                requestTokenArn = requestToken
            },
        )
        status = describeResponse.asyncOperation?.requestStatus
        println(status)
    }
}

```

- Weitere Informationen finden Sie unter [AWS SDK für den Kotlin-Entwicklerhandbuch](#).

- API-Einheiten finden Sie in [CreateMultiRegionAccessPoint](#) AWS SDK als API Kotlin-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **CreateMultipartUpload** mit einem AWS SDK oder CLI


Die folgenden Codebeispiele zeigen die Verwendung von `CreateMultipartUpload`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erstellen einer mehrteiligen Kopie](#)
- [Durchführen eines mehrteiligen Uploads](#)
- [Verwenden der Prüfsummen](#)
- [Arbeiten Sie mit Amazon S3 S3-Objektintegrität](#)

C++

SDK für C++

 Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#!/ Create a multipart upload.
/*!
  \param bucket: The name of the S3 bucket where the object will be uploaded.
  \param key: The unique identifier (key) for the object within the S3 bucket.
  \param client: The S3 client instance used to perform the upload operation.
  \return Aws::String: Upload ID or empty string if failed.
*/
Aws::String
AwsDoc::S3::createMultipartUpload(const Aws::String &bucket, const Aws::String
&key,
```

```

        Aws::S3::Model::ChecksumAlgorithm
checksumAlgorithm,
        const Aws::S3::S3Client &client) {
    Aws::S3::Model::CreateMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);

    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }

    Aws::S3::Model::CreateMultipartUploadOutcome outcome =
        client.CreateMultipartUpload(request);

    Aws::String uploadID;
    if (outcome.IsSuccess()) {
        uploadID = outcome.GetResult().GetUploadId();
    } else {
        std::cerr << "Error creating multipart upload: " <<
outcome.GetError().GetMessage() << std::endl;
    }

    return uploadID;
}

```

- API-Einheiten finden Sie [CreateMultipartUpload](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl erstellt einen mehrteiligen Upload im Bucket `my-bucket` mit dem Schlüssel `multipart/01`:

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Ausgabe:

```
{
  "Bucket": "my-bucket",
```

```

    "UploadId":
      "dfRtDYU0WCCcH43C3WfbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3
      "Key": "multipart/01"
  }

```

Die fertige Datei wird 01 in einem Ordner namens multipart Bucket my-bucket benannt. Speichern Sie die Upload-ID, den Schlüssel und den Bucket-Namen zur Verwendung mit dem upload-part Befehl.

- API-Einheiten finden Sie [CreateMultipartUpload](#) unter AWS CLI Befehlsreferenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

// Create a multipart upload. Use UploadPart and CompleteMultipartUpload to
// upload the file.
let multipart_upload_res: CreateMultipartUploadOutput = client
    .create_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .send()
    .await?;

let upload_id = multipart_upload_res.upload_id().ok_or(S3ExampleError::new(
    "Missing upload_id after CreateMultipartUpload",
))?;

```

```

let mut upload_parts: Vec<aws_sdk_s3::types::CompletedPart> = Vec::new();

for chunk_index in 0..chunk_count {
    let this_chunk = if chunk_count - 1 == chunk_index {
        size_of_last_chunk
    } else {

```



```
        CHUNK_SIZE
    };
    let stream = ByteStream::read_from()
        .path(path)
        .offset(chunk_index * CHUNK_SIZE)
        .length(Length::Exact(this_chunk))
        .build()
        .await
        .unwrap();

    // Chunk index needs to start at 0, but part numbers start at 1.
    let part_number = (chunk_index as i32) + 1;
    let upload_part_res = client
        .upload_part()
        .key(&key)
        .bucket(&bucket_name)
        .upload_id(upload_id)
        .body(stream)
        .part_number(part_number)
        .send()
        .await?;

    upload_parts.push(
        CompletedPart::builder()
            .e_tag(upload_part_res.e_tag.unwrap_or_default())
            .part_number(part_number)
            .build(),
    );
}
```

```
// upload_parts: Vec<aws_sdk_s3::types::CompletedPart>
let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
    .set_parts(Some(upload_parts))
    .build();

let _complete_multipart_upload_res = client
    .complete_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .multipart_upload(completed_multipart_upload)
    .upload_id(upload_id)
```

```
.send()  
.await?;
```

- API-Einheiten finden Sie [CreateMultipartUpload](#) unter AWS SDK als API-Referenz auf Rust.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucket** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `DeleteBucket`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erlernen Sie die Grundlagen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>  
/// Shows how to delete an Amazon S3 bucket.  
/// </summary>  
/// <param name="client">An initialized Amazon S3 client object.</param>  
/// <param name="bucketName">The name of the Amazon S3 bucket to  
delete.</param>  
/// <returns>A boolean value that represents the success or failure of  
/// the delete operation.</returns>  
public static async Task<bool> DeleteBucketAsync(IAmazonS3 client, string  
bucketName)
```

```

    {
        var request = new DeleteBucketRequest
        {
            BucketName = bucketName,
        };

        var response = await client.DeleteBucketAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }

```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:

```

```
#      0 - If successful.
#      1 - If it fails.
#####
function delete_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api delete-bucket \
        --bucket "$bucket_name")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
        return 1
    fi
}
}
```

- API Einzelheiten finden Sie [DeleteBucket](#) unter AWS CLI Befehlsreferenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::deleteBucket(const Aws::String &bucketName,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::DeleteBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketOutcome outcome =
        client.DeleteBucket(request);
}
```

```
    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: deleteBucket: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        std::cout << "The bucket was deleted" << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl löscht einen Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- API-Einheiten finden Sie unter [DeleteBucket](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
bucket
```

```
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// DeleteBucket deletes a bucket. The bucket must be empty or an error is
// returned.
func (basics BucketBasics) DeleteBucket(bucketName string) error {
    _, err := basics.S3Client.DeleteBucket(context.TODO(), &s3.DeleteBucketInput{
        Bucket: aws.String(bucketName)})
    if err != nil {
        log.Printf("Couldn't delete bucket %v. Here's why: %v\n", bucketName, err)
    }
    return err
}
```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/**
 * Deletes an S3 bucket asynchronously.
 *
 * @param bucket the name of the bucket to be deleted
 * @return a {@link CompletableFuture} that completes when the bucket
 * deletion is successful, or throws a {@link RuntimeException}
 *         if an error occurs during the deletion process
 */
public CompletableFuture<Void> deleteBucketAsync(String bucket) {
    DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
```

```
        .bucket(bucket)
        .build();

        CompletableFuture<DeleteBucketResponse> response =
getAsyncClient().deleteBucket(deleteBucketRequest);
        response.whenComplete((deleteRes, ex) -> {
            if (deleteRes != null) {
                logger.info(bucket + " was deleted.");
            } else {
                throw new RuntimeException("An S3 exception occurred during
bucket deletion", ex);
            }
        });
        return response.thenApply(r -> null);
    }
}
```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie den Bucket.

```
import { DeleteBucketCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Delete a bucket.
export const main = async () => {
    const command = new DeleteBucketCommand({
        Bucket: "test-bucket",
    });

    try {
```

```
const response = await client.send(command);
console.log(response);
} catch (err) {
  console.error(err);
}
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [DeleteBucket](#) in AWS SDK for JavaScript API-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie einen leeren Bucket.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
  $this->s3client->deleteBucket([
    'Bucket' => $this->bucketName,
  ]);
  echo "Deleted bucket $this->bucketName.\n";
} catch (Exception $exception) {
  echo "Failed to delete $this->bucketName with error: " . $exception-
  >getMessage();
  exit("Please fix error with bucket deletion before continuing.");
}
```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl entfernt alle Objekte und Objektversionen aus dem Bucket 'test-files' und löscht dann den Bucket. Der Befehl fordert Sie zur Bestätigung auf, bevor Sie fortfahren. Fügen Sie den Schalter `-Force` hinzu, um die Bestätigung zu unterdrücken. Beachten Sie, dass Buckets, die nicht leer sind, nicht gelöscht werden können.

```
Remove-S3Bucket -BucketName test-files -DeleteBucketContent
```

- API-Einheiten finden Sie unter [DeleteBucket](#) AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def delete(self):
        """
        Delete the bucket. The bucket must be empty or an error is raised.
        """
        try:
```

```
self.bucket.delete()
self.bucket.wait_until_not_exists()
logger.info("Bucket %s successfully deleted.", self.bucket.name)
except ClientError:
    logger.exception("Couldn't delete bucket %s.", self.bucket.name)
    raise
```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?")
  answer = gets.chomp.downcase
  if answer == "y"
    bucket.objects.batch_delete!
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
pub async fn delete_bucket(
    client: &aws_sdk_s3::Client,
    bucket_name: &str,
) -> Result<(), S3ExampleError> {
    let resp = client.delete_bucket().bucket(bucket_name).send().await;
    match resp {
        Ok(_) => Ok(()),
        Err(err) => {
            if err
                .as_service_error()
                .and_then(aws_sdk_s3::error::ProvideErrorMetadata::code)
                == Some("NoSuchBucket")
            {
                Ok(())
            } else {
                Err(S3ExampleError::from(err))
            }
        }
    }
}
```

- API Einheiten finden Sie [DeleteBucket](#) unter AWS SDK als API Referenz auf Rust.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
TRY.  
  
    lo_s3->deletebucket(  
        iv_bucket = iv_bucket_name  
    ).  
    MESSAGE 'Deleted S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- API Einzelheiten finden Sie [DeleteBucket](#) unter AWS SDK als SAP ABAP API Referenz.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3  
  
public func deleteBucket(name: String) async throws {  
    let input = DeleteBucketInput(  
        bucket: name  
    )  
    do {
```

```
        _ = try await client.deleteBucket(input: input)
    }
    catch {
        print("ERROR: ", dump(error, name: "Deleting a bucket"))
        throw error
    }
}
```

- API-Einheiten finden Sie [DeleteBucket](#) unter AWS SDK als API-Swift-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucketAnalyticsConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von **DeleteBucketAnalyticsConfiguration**.

CLI

AWS CLI

Um eine Analytics-Konfiguration für einen Bucket zu löschen

Im folgenden `delete-bucket-analytics-configuration`-Beispiel wird die Analytics-Konfiguration für den angegebenen Bucket und die angegebene ID entfernt.

```
aws s3api delete-bucket-analytics-configuration \
  --bucket my-bucket \
  --id 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteBucketAnalyticsConfiguration](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl entfernt den Analysefilter mit dem Namen 'testfilter' aus dem angegebenen S3-Bucket.

```
Remove-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId  
'testfilter'
```

- API-Einheiten finden Sie unter [DeleteBucketAnalyticsConfiguration](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucketCors** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `DeleteBucketCors`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>  
/// Deletes a CORS configuration from an Amazon S3 bucket.  
/// </summary>  
/// <param name="client">The initialized Amazon S3 client object used  
/// to delete the CORS configuration from the bucket.</param>  
private static async Task DeleteCORSConfigurationAsync(AmazonS3Client  
client)  
{
```

```
        DeleteCORSConfigurationRequest request = new
DeleteCORSConfigurationRequest()
    {
        BucketName = BucketName,
    };
    await client.DeleteCORSConfigurationAsync(request);
}
```

- API-Einheiten finden Sie [DeleteBucketCors](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Der folgende Befehl löscht eine Cross-Origin Resource Sharing-Konfiguration aus einem Bucket mit dem Namen: `my-bucket`

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- API-Einheiten finden Sie unter [DeleteBucketCors](#) AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
```

```
        that wraps bucket actions in a class-like structure.
    """
    self.bucket = bucket
    self.name = bucket.name

    def delete_cors(self):
        """
        Delete the CORS rules from the bucket.

        :param bucket_name: The name of the bucket to update.
        """
        try:
            self.bucket.Cors().delete()
            logger.info("Deleted CORS from bucket '%s'.", self.bucket.name)
        except ClientError:
            logger.exception("Couldn't delete CORS from bucket '%s'.",
                self.bucket.name)
            raise
```

- API Einzelheiten finden Sie [DeleteBucketCors](#) unter AWS SDK für Python (Boto3) API - Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors
```



```
# @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
an existing bucket.
def initialize(bucket_cors)
  @bucket_cors = bucket_cors
end

# Deletes the CORS configuration of a bucket.
#
# @return [Boolean] True if the CORS rules were deleted; otherwise, false.
def delete_cors
  @bucket_cors.delete
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't delete CORS rules for #{@bucket_cors.bucket.name}. Here's why:
#{e.message}"
  false
end

end
```

- API Einzelheiten finden Sie [DeleteBucketCors](#) unter AWS SDK for Ruby API Referenz.

Für eine vollständige Liste von AWS SDK Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **DeleteBucketEncryption** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `DeleteBucketEncryption`.

CLI

AWS CLI

Um die serverseitige Verschlüsselungskonfiguration eines Buckets zu löschen

Im folgenden `delete-bucket-encryption` Beispiel wird die serverseitige Verschlüsselungskonfiguration des angegebenen Buckets gelöscht.

```
aws s3api delete-bucket-encryption \  
  --bucket my-bucket
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API Einzelheiten finden Sie unter [DeleteBucketEncryption](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dadurch wird die für den bereitgestellten S3-Bucket aktivierte Verschlüsselung deaktiviert.

```
Remove-S3BucketEncryption -BucketName 's3casetestbucket'
```

Ausgabe:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketEncryption (DeleteBucketEncryption)" on
target "s3casetestbucket".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
```

- API Einzelheiten finden Sie unter [DeleteBucketEncryption](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **DeleteBucketInventoryConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `DeleteBucketInventoryConfiguration`.

CLI

AWS CLI

Um die Inventarkonfiguration eines Buckets zu löschen

Im folgenden `delete-bucket-inventory-configuration` Beispiel wird die Inventarkonfiguration mit der ID 1 für den angegebenen Bucket gelöscht.

```
aws s3api delete-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteBucketInventoryConfiguration](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl entfernt das Inventar mit dem Namen 'testInventoryName', das dem angegebenen S3-Bucket entspricht.

```
Remove-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId  
'testInventoryName'
```

Ausgabe:

```
Confirm  
Are you sure you want to perform this action?  
Performing the operation "Remove-S3BucketInventoryConfiguration  
(DeleteBucketInventoryConfiguration)" on target "s3testbucket".  
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is  
"Y"): Y
```

- API-Einheiten finden Sie unter [DeleteBucketInventoryConfiguration](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucketLifecycle** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `DeleteBucketLifecycle`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// This method removes the Lifecycle configuration from the named
/// S3 bucket.
/// </summary>
/// <param name="client">The S3 client object used to call
/// the RemoveLifecycleConfigAsync method.</param>
/// <param name="bucketName">A string representing the name of the
/// S3 bucket from which the configuration will be removed.</param>
public static async Task RemoveLifecycleConfigAsync(IAmazonS3 client,
string bucketName)
{
    var request = new DeleteLifecycleConfigurationRequest()
    {
        BucketName = bucketName,
    };
    await client.DeleteLifecycleConfigurationAsync(request);
}
```

- API-Einheiten finden Sie [DeleteBucketLifecycle](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Der folgende Befehl löscht eine Lebenszykluskonfiguration aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- API-Einheiten finden Sie unter [DeleteBucketLifecycle](#) AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def delete_lifecycle_configuration(self):
        """
        Remove the lifecycle configuration from the specified bucket.
        """
        try:
            self.bucket.LifecycleConfiguration().delete()
            logger.info(
```

```
        "Deleted lifecycle configuration for bucket '%s'.",
self.bucket.name
    )
    except ClientError:
        logger.exception(
            "Couldn't delete lifecycle configuration for bucket '%s'.",
            self.bucket.name,
        )
        raise
```

- API-Einheiten finden Sie [DeleteBucketLifecycle](#) unter AWS SDK für Python (Boto3) API - Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucketMetricsConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `DeleteBucketMetricsConfiguration`.

CLI

AWS CLI

Um eine Metrikkonfiguration für einen Bucket zu löschen

Im folgenden `delete-bucket-metrics-configuration` Beispiel wird die Metrikkonfiguration für den angegebenen Bucket und die angegebene ID entfernt.

```
aws s3api delete-bucket-metrics-configuration \
  --bucket my-bucket \
  --id 123
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteBucketMetricsConfiguration](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl entfernt den Metrikfilter mit dem Namen 'testmetrics' im angegebenen S3-Bucket.

```
Remove-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId  
'testmetrics'
```

- API-Einheiten finden Sie unter [DeleteBucketMetricsConfiguration](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucketPolicy** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `DeleteBucketPolicy`.

C++

SDK für C++

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::deleteBucketPolicy(const Aws::String &bucketName,  
                                     const Aws::S3::S3ClientConfiguration  
&clientConfig) {  
    Aws::S3::S3Client client(clientConfig);  
  
    Aws::S3::Model::DeleteBucketPolicyRequest request;  
    request.SetBucket(bucketName);  
  
    Aws::S3::Model::DeleteBucketPolicyOutcome outcome =  
    client.DeleteBucketPolicy(request);
```

```
if (!outcome.IsSuccess()) {
    const Aws::S3::S3Error &err = outcome.GetError();
    std::cerr << "Error: deleteBucketPolicy: " <<
        err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
} else {
    std::cout << "Policy was deleted from the bucket." << std::endl;
}

return outcome.IsSuccess();
}
```

- API-Einheiten finden Sie [DeleteBucketPolicy](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl löscht eine Bucket-Richtlinie aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- API-Einheiten finden Sie unter [DeleteBucketPolicy](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
```



```
import software.amazon.awssdk.services.s3.model.DeleteBucketPolicyRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class DeleteBucketPolicy {
    public static void main(String[] args) {

        final String usage = ""

            Usage:
                <bucketName>

            Where:
                bucketName - The Amazon S3 bucket to delete the policy from
(for example, bucket1)."";

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        System.out.format("Deleting policy from bucket: \"%s\"\n\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        deleteS3BucketPolicy(s3, bucketName);
        s3.close();
    }

    // Delete the bucket policy.
    public static void deleteS3BucketPolicy(S3Client s3, String bucketName) {
        DeleteBucketPolicyRequest delReq = DeleteBucketPolicyRequest.builder()
            .bucket(bucketName)
```

```
        .build();

    try {
        s3.deleteBucketPolicy(delReq);
        System.out.println("Done!");
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- API-Einheiten finden Sie [DeleteBucketPolicy](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie die Bucket-Richtlinie.

```
import { DeleteBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// This will remove the policy from the bucket.
export const main = async () => {
    const command = new DeleteBucketPolicyCommand({
        Bucket: "test-bucket",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
```

```
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [DeleteBucketPolicy](#) in AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun deleteS3BucketPolicy(bucketName: String?) {
    val request =
        DeleteBucketPolicyRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteBucketPolicy(request)
        println("Done!")
    }
}
```

- API-Einheiten finden Sie [DeleteBucketPolicy](#) unter AWS SDK als API Kotlin-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl entfernt die Bucket-Richtlinie, die dem angegebenen S3-Bucket zugeordnet ist.

```
Remove-S3BucketPolicy -BucketName 's3testbucket'
```

- API-Einheiten finden Sie [DeleteBucketPolicy](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def delete_policy(self):
        """
        Delete the security policy from the bucket.
        """
        try:
            self.bucket.Policy().delete()
            logger.info("Deleted policy for bucket '%s'.", self.bucket.name)
        except ClientError:
            logger.exception(
                "Couldn't delete policy for bucket '%s'.", self.bucket.name
            )
            raise
```

- API-Einheiten finden Sie [DeleteBucketPolicy](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object
  # configured with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  def delete_policy
    @bucket_policy.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete the policy from #{@bucket_policy.bucket.name}. Here's
  why: #{e.message}"
    false
  end
end

end
```

- API-Einheiten finden Sie [DeleteBucketPolicy](#) unter AWS SDK for Ruby API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucketReplication** mit einem AWS SDK oder der CLI.

Die folgenden Codebeispiele zeigen die Verwendung von `DeleteBucketReplication`.

CLI

AWS CLI

Der folgende Befehl löscht eine Replikationskonfiguration aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- API-Einheiten finden Sie unter [DeleteBucketReplication](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Löscht die Replikationskonfiguration, die dem Bucket mit dem Namen „mybucket“ zugeordnet ist. Beachten Sie, dass für diesen Vorgang eine Genehmigung für die Aktion `s3:DeleteReplicationConfiguration` erforderlich ist. Sie werden zur Bestätigung aufgefordert, bevor der Vorgang fortgesetzt wird. Um die Bestätigung zu unterdrücken, verwenden Sie den Schalter `-Force`.

```
Remove-S3BucketReplication -BucketName mybucket
```

- API-Einheiten finden Sie unter [DeleteBucketReplication](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteBucketTagging** mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen die VerwendungDeleteBucketTagging.

CLI

AWS CLI

Der folgende Befehl löscht eine Tagging-Konfiguration aus einem Bucket mit dem Namen: my-bucket

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- APIEinzelheiten finden Sie unter [DeleteBucketTagging](#)AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl entfernt alle Tags, die dem angegebenen S3-Bucket zugeordnet sind.

```
Remove-S3BucketTagging -BucketName 's3testbucket'
```

Ausgabe:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketTagging (DeleteBucketTagging)" on target
"s3testbucket".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
```

- APIEinzelheiten finden Sie [DeleteBucketTagging](#)unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDKEntwicklerhandbücher und Codebeispiele finden Sie unter[Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **DeleteBucketWebsite** mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen die Verwendung `DeleteBucketWebsite`.

C++

SDKfür C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::deleteBucketWebsite(const Aws::String &bucketName,
                                     const Aws::S3::S3ClientConfiguration
                                     &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::DeleteBucketWebsiteRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketWebsiteOutcome outcome =
        client.DeleteBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error: deleteBucketWebsite: " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        std::cout << "Website configuration was removed." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- APIEinzelheiten finden Sie [DeleteBucketWebsite](#)unter AWS SDK for C++ APIReferenz.

CLI

AWS CLI

Der folgende Befehl löscht eine Website-Konfiguration aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-website --bucket my-bucket
```

- API-Einheiten finden Sie unter [DeleteBucketWebsite](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.DeleteBucketWebsiteRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */

public class DeleteWebsiteConfiguration {
    public static void main(String[] args) {
        final String usage = ""

        Usage:      <bucketName>
```

```
        Where:
            bucketName - The Amazon S3 bucket to delete the website
configuration from.
        """;

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String bucketName = args[0];
    System.out.format("Deleting website configuration for Amazon S3 bucket:
%s\n", bucketName);
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    deleteBucketWebsiteConfig(s3, bucketName);
    System.out.println("Done!");
    s3.close();
}

public static void deleteBucketWebsiteConfig(S3Client s3, String bucketName)
{
    DeleteBucketWebsiteRequest delReq = DeleteBucketWebsiteRequest.builder()
        .bucket(bucketName)
        .build();

    try {
        s3.deleteBucketWebsite(delReq);
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.out.println("Failed to delete website configuration!");
        System.exit(1);
    }
}
}
```

- API-Einheiten finden Sie [DeleteBucketWebsite](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie die Website-Konfiguration aus dem Bucket.

```
import { DeleteBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Disable static website hosting on the bucket.
export const main = async () => {
  const command = new DeleteBucketWebsiteCommand({
    Bucket: "test-bucket",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [DeleteBucketWebsite](#) in AWS SDK for JavaScript API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl deaktiviert die statische Website-Hosting-Eigenschaft des angegebenen S3-Buckets.

```
Remove-S3BucketWebsite -BucketName 's3testbucket'
```

Ausgabe:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketWebsite (DeleteBucketWebsite)" on target
"s3testbucket".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
```

- API-Einheiten finden Sie unter [DeleteBucketWebsite](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteObject** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `DeleteObject`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Arbeiten Sie mit Amazon S3 S3-Objektintegrität](#)
- [Mit versionierten Objekten arbeiten](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie ein Objekt in einem nicht versionierten S3-Bucket.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to delete an object from a non-versioned Amazon
/// Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class DeleteObject
{
    /// <summary>
    /// The Main method initializes the necessary variables and then calls
    /// the DeleteObjectNonVersionedBucketAsync method to delete the object
    /// named by the keyName parameter.
    /// </summary>
    public static async Task Main()
    {
        const string bucketName = "amzn-s3-demo-bucket";
        const string keyName = "testfile.txt";

        // If the Amazon S3 bucket is located in an AWS Region other than the
        // Region of the default account, define the AWS Region for the
        // Amazon S3 bucket in your call to the AmazonS3Client constructor.
        // For example RegionEndpoint.USWest2.
        IAmazonS3 client = new AmazonS3Client();
        await DeleteObjectNonVersionedBucketAsync(client, bucketName,
keyName);
    }

    /// <summary>
    /// The DeleteObjectNonVersionedBucketAsync takes care of deleting the
    /// desired object from the named bucket.
    /// </summary>
    /// <param name="client">An initialized Amazon S3 client used to delete
    /// an object from an Amazon S3 bucket.</param>
    /// <param name="bucketName">The name of the bucket from which the
    /// object will be deleted.</param>
    /// <param name="keyName">The name of the object to delete.</param>
    public static async Task DeleteObjectNonVersionedBucketAsync(IAmazonS3
client, string bucketName, string keyName)
    {
        try
```

```
        {
            var deleteObjectRequest = new DeleteObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
            };

            Console.WriteLine($"Deleting object: {keyName}");
            await client.DeleteObjectAsync(deleteObjectRequest);
            Console.WriteLine($"Object: {keyName} deleted from
{bucketName}.");
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error encountered on server.
Message: '{ex.Message}' when deleting an object.");
        }
    }
}
```

Löschen Sie ein Objekt in einem versionierten S3-Bucket.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example creates an object in an Amazon Simple Storage Service
/// (Amazon S3) bucket and then deletes the object version that was
/// created.
/// </summary>
public class DeleteObjectVersion
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";
        string keyName = "verstioned-object.txt";

        // If the AWS Region of the default user is different from the AWS
        // Region of the Amazon S3 bucket, pass the AWS Region of the
```

```
        // bucket region to the Amazon S3 client object's constructor.
        // Define it like this:
        //     RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        IAmazonS3 client = new AmazonS3Client();

        await CreateAndDeleteObjectVersionAsync(client, bucketName, keyName);
    }

    /// <summary>
    /// This method creates and then deletes a versioned object.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
    /// create and delete the object.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket where the
    /// object will be created and deleted.</param>
    /// <param name="keyName">The key name of the object to create.</param>
    public static async Task CreateAndDeleteObjectVersionAsync(IAmazonS3
client, string bucketName, string keyName)
    {
        try
        {
            // Add a sample object.
            string versionID = await PutAnObject(client, bucketName,
keyName);

            // Delete the object by specifying an object key and a version
ID.

            DeleteObjectRequest request = new DeleteObjectRequest()
            {
                BucketName = bucketName,
                Key = keyName,
                VersionId = versionID,
            };

            Console.WriteLine("Deleting an object");
            await client.DeleteObjectAsync(request);
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }

    /// <summary>
```

```

    /// This method is used to create the temporary Amazon S3 object.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 object which will be
used
    /// to create the temporary Amazon S3 object.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket where the
object
    /// will be created.</param>
    /// <param name="objectKey">The name of the Amazon S3 object co create.</
param>
    /// <returns>The Version ID of the created object.</returns>
    public static async Task<string> PutAnObject(IAmazonS3 client, string
bucketName, string objectKey)
    {
        PutObjectRequest request = new PutObjectRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            ContentBody = "This is the content body!",
        };

        PutObjectResponse response = await client.PutObjectAsync(request);
        return response.VersionId;
    }
}

```

- API-Einheiten finden Sie [DeleteObject](#) unter AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

#####
# function errecho

```



```

#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_item_in_bucket
#
# This function deletes the specified file from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - The key (file name) in the bucket to delete.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_item_in_bucket() {
    local bucket_name=$1
    local key=$2
    local response

    response=$(aws s3api delete-object \
        --bucket "$bucket_name" \
        --key "$key")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n
$response"
        return 1
    fi
}

```

- API-Einheiten finden Sie [DeleteObject](#) unter AWS CLI Befehlsreferenz.

C++

SDK für C++

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::deleteObject(const Aws::String &objectKey,
                              const Aws::String &fromBucket,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::DeleteObjectRequest request;

    request.WithKey(objectKey)
           .WithBucket(fromBucket);

    Aws::S3::Model::DeleteObjectOutcome outcome =
        client.DeleteObject(request);

    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error: deleteObject: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        std::cout << "Successfully deleted the object." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API Einzelheiten finden Sie [DeleteObject](#) unter AWS SDK for C++ API Referenz.

CLI

AWS CLI

Der folgende Befehl löscht ein Objekt mit dem Namen `test.txt` aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Wenn die Bucket-Versionierung aktiviert ist, enthält die Ausgabe die Versions-ID der Löschmarkierung:

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tND1W1Gq",
  "DeleteMarker": true
}
```

Weitere Informationen zum Löschen von Objekten finden Sie unter [Objekte löschen](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie [DeleteObject](#) unter AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu [GitHub](#). Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
  S3Client *s3.Client
  S3Manager *manager.Uploader
}
```

```
// DeleteObject deletes an object from a bucket.
func (actor S3Actions) DeleteObject(ctx context.Context, bucket string, key
string, versionId string, bypassGovernance bool) (bool, error) {
    deleted := false
    input := &s3.DeleteObjectInput{
        Bucket: aws.String(bucket),
        Key:     aws.String(key),
    }
    if versionId != "" {
        input.VersionId = aws.String(versionId)
    }
    if bypassGovernance {
        input.BypassGovernanceRetention = aws.Bool(true)
    }
    _, err := actor.S3Client.DeleteObject(ctx, input)
    if err != nil {
        var noKey *types.NoSuchKey
        var apiErr *smithy.GenericAPIError
        if errors.As(err, &noKey) {
            log.Printf("Object %s does not exist in %s.\n", key, bucket)
            err = noKey
        } else if errors.As(err, &apiErr) {
            switch apiErr.ErrorCode() {
            case "AccessDenied":
                log.Printf("Access denied: cannot delete object %s from %s.\n", key, bucket)
                err = nil
            case "InvalidArgument":
                if bypassGovernance {
                    log.Printf("You cannot specify bypass governance on a bucket without lock
enabled.")
                    err = nil
                }
            }
        }
    } else {
        deleted = true
    }
    return deleted, err
}
```

- APIEinzelheiten finden Sie [DeleteObject](#) unter AWS SDK for Go APIReferenz.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/**
 * Deletes an object from an S3 bucket asynchronously.
 *
 * @param bucketName the name of the S3 bucket
 * @param key the key (file name) of the object to be deleted
 * @return a {@link CompletableFuture} that completes when the object has
 * been deleted
 */
public CompletableFuture<Void> deleteObjectFromBucketAsync(String bucketName,
String key) {
    DeleteObjectRequest deleteObjectRequest = DeleteObjectRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();

    CompletableFuture<DeleteObjectResponse> response =
getAsyncClient().deleteObject(deleteObjectRequest);
    response.whenComplete((deleteRes, ex) -> {
        if (deleteRes != null) {
            logger.info(key + " was deleted");
        } else {
            throw new RuntimeException("An S3 exception occurred during
delete", ex);
        }
    });

    return response.thenApply(r -> null);
}
```

- API Einzelheiten finden Sie [DeleteObject](#) unter AWS SDK for Java 2.x API Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie ein Objekt.

```
import { DeleteObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new DeleteObjectCommand({
    Bucket: "test-bucket",
    Key: "test-key.txt",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- API-Einheiten finden Sie [DeleteObject](#) unter AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie ein Objekt.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
                               that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def delete(self):
        """
        Deletes the object.
        """
        try:
            self.object.delete()
            self.object.wait_until_not_exists()
            logger.info(
                "Deleted object '%s' from bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't delete object '%s' from bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
```

```
raise
```

Setzen Sie ein Objekt auf eine vorherige Version zurück, indem Sie spätere Versionen des Objekts löschen.

```
def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(
        bucket.object_versions.filter(Prefix=object_key),
        key=attrgetter("last_modified"),
        reverse=True,
    )

    logger.debug(
        "Got versions:\n%s",
        "\n".join(
            [
                f"\t{version.version_id}, last modified {version.last_modified}"
                for version in versions
            ]
        ),
    )

    if version_id in [ver.version_id for ver in versions]:
        print(f"Rolling back to version {version_id}")
        for version in versions:
            if version.version_id != version_id:
                version.delete()
                print(f"Deleted version {version.version_id}")
```



```
        else:
            break

        print(f"Active version is now {bucket.Object(object_key).version_id}")
    else:
        raise KeyError(
            f"{version_id} was not found in the list of versions for "
            f"{object_key}."
        )
```

Reaktivieren Sie ein gelöschttes Objekt, indem Sie die aktive Löschmarke des Objekts entfernen.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
    marker.
    By removing the delete marker, we make the previous version the latest
    version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1
    )

    if "DeleteMarkers" in response:
        latest_version = response["DeleteMarkers"][0]
        if latest_version["IsLatest"]:
            logger.info(
                "Object %s was indeed deleted on %s. Let's revive it.",
                object_key,
```

```
        latest_version["LastModified"],
    )
    obj = bucket.Object(object_key)
    obj.Version(latest_version["VersionId"]).delete()
    logger.info(
        "Revived %s, active version is now %s with body '%s'",
        object_key,
        obj.version_id,
        obj.get()["Body"].read(),
    )
else:
    logger.warning(
        "Delete marker is not the latest version for %s!", object_key
    )
elif "Versions" in response:
    logger.warning("Got an active version for %s, nothing to do.",
object_key)
else:
    logger.error("Couldn't get any version info for %s.", object_key)
```

Erstellen Sie einen Lambda-Handler, der eine Löschmarke aus einem S3-Objekt entfernt. Dieser Handler kann verwendet werden, um irrelevante Löschmarkierungen in einem versionierten Bucket effizient zu bereinigen.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.client("s3")

def lambda_handler(event, context):
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
```

```
        to remove.
:param context: Context about the event.
:return: A result structure that Amazon S3 uses to interpret the result of
the
        operation. When the result code is TemporaryFailure, S3 retries the
        operation.
"""
# Parse job parameters from Amazon S3 batch operations
invocation_id = event["invocationId"]
invocation_schema_version = event["invocationSchemaVersion"]

results = []
result_code = None
result_string = None

task = event["tasks"][0]
task_id = task["taskId"]

try:
    obj_key = parse.unquote(task["s3Key"], encoding="utf-8")
    obj_version_id = task["s3VersionId"]
    bucket_name = task["s3BucketArn"].split(":")[-1]

    logger.info(
obj_key
        "Got task: remove delete marker %s from object %s.", obj_version_id,
    )

    try:
delete
        # If this call does not raise an error, the object version is not a
        # marker and should not be deleted.
        response = s3.head_object(
            Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
        )
        result_code = "PermanentFailure"
        result_string = (
marker."
            f"Object {obj_key}, ID {obj_version_id} is not " f"a delete
        )

        logger.debug(response)
        logger.warning(result_string)
    except ClientError as error:
```

```
        delete_marker = error.response["ResponseMetadata"]
["HTTPHeaders"].get(
            "x-amz-delete-marker", "false"
        )
        if delete_marker == "true":
            logger.info(
                "Object %s, version %s is a delete marker.", obj_key,
obj_version_id
            )
            try:
                s3.delete_object(
                    Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
                )
                result_code = "Succeeded"
                result_string = (
                    f"Successfully removed delete marker "
                    f"{obj_version_id} from object {obj_key}."
                )
                logger.info(result_string)
            except ClientError as error:
                # Mark request timeout as a temporary failure so it will be
retried.

                if error.response["Error"]["Code"] == "RequestTimeout":
                    result_code = "TemporaryFailure"
                    result_string = (
                        f"Attempt to remove delete marker from "
                        f"object {obj_key} timed out."
                    )
                    logger.info(result_string)
                else:
                    raise
            else:
                raise ValueError(
                    f"The x-amz-delete-marker header is either not "
                    f"present or is not 'true'."
                )
        except Exception as error:
            # Mark all other exceptions as permanent failures.
            result_code = "PermanentFailure"
            result_string = str(error)
            logger.exception(error)
    finally:
        results.append(
            {
```

```
        "taskId": task_id,  
        "resultCode": result_code,  
        "resultString": result_string,  
    }  
)  
return {  
    "invocationSchemaVersion": invocation_schema_version,  
    "treatMissingKeysAs": "PermanentFailure",  
    "invocationId": invocation_id,  
    "results": results,  
}
```

- API-Einheiten finden Sie [DeleteObject](#) unter AWS SDK für Python (Boto3) API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// Delete an object from a bucket.  
pub async fn remove_object(  
    client: &aws_sdk_s3::Client,  
    bucket: &str,  
    key: &str,  
) -> Result<(), S3ExampleError> {  
    client  
        .delete_object()  
        .bucket(bucket)  
        .key(key)  
        .send()  
        .await?;  
  
    // There are no modeled errors to handle when deleting an object.
```

```
Ok(())  
}
```

- API-Einheiten finden Sie [DeleteObject](#) unter AWS SDK als API-Referenz auf Rust.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
TRY.  
  lo_s3->deleteobject(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_object_key  
  ).  
  MESSAGE 'Object deleted from S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- API-Einheiten finden Sie [DeleteObject](#) unter AWS SDK als SAP ABAP API-Referenz.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3

public func deleteFile(bucket: String, key: String) async throws {
    let input = DeleteObjectInput(
        bucket: bucket,
        key: key
    )

    do {
        _ = try await client.deleteObject(input: input)
    }
    catch {
        print("ERROR: ", dump(error, name: "Deleting a file. "))
        throw error
    }
}
```

- API Einzelheiten finden Sie [DeleteObject](#) unter AWS SDK als API Swift-Referenz.

Für eine vollständige Liste von AWS SDK Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **DeleteObjectTagging** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `DeleteObjectTagging`.

CLI

AWS CLI

Um die Tag-Sets eines Objekts zu löschen

Im folgenden `delete-object-tagging` Beispiel wird das Tag mit dem angegebenen Schlüssel aus dem Objekt `doc1.rtf` gelöscht.

```
aws s3api delete-object-tagging \
  --bucket my-bucket \
  --key doc1.rtf
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteObjectTagging](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl entfernt alle Tags, die dem Objekt mit dem Schlüssel 'testfile.txt' im angegebenen S3-Bucket zugeordnet sind.

```
Remove-S3ObjectTagSet -Key 'testfile.txt' -BucketName 's3testbucket' -Select '^Key'
```

Ausgabe:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3ObjectTagSet (DeleteObjectTagging)" on target
"testfile.txt".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
testfile.txt
```

- API-Einheiten finden Sie [DeleteObjectTagging](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteObjects** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `DeleteObjects`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erlernen Sie die Grundlagen](#)
- [Löscht alle Objekte in einem Bucket](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie alle Objekte aus einem S3 Bucket.

```
/// <summary>
/// Delete all of the objects stored in an existing Amazon S3 bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket from which the
/// contents will be deleted.</param>
/// <returns>A boolean value that represents the success or failure of
/// deleting all of the objects in the bucket.</returns>
public static async Task<bool> DeleteBucketContentsAsync(IAmazonS3
client, string bucketName)
{
    // Iterate over the contents of the bucket and delete all objects.
    var request = new ListObjectsV2Request
    {
        BucketName = bucketName,
    };

    try
    {
        ListObjectsV2Response response;

        do
        {
            response = await client.ListObjectsV2Async(request);
            response.S3Objects
                .ForEach(async obj => await
client.DeleteObjectAsync(bucketName, obj.Key));

            // If the response is truncated, set the request
ContinuationToken
```

```

        // from the NextContinuationToken property of the response.
        request.ContinuationToken = response.NextContinuationToken;
    }
    while (response.IsTruncated);

    return true;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error deleting objects: {ex.Message}");
    return false;
}
}

```

Löschen Sie mehrere Objekte in einem nicht versionierten S3-Bucket.

```

using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to delete multiple objects from an Amazon Simple
/// Storage Service (Amazon S3) bucket.
/// </summary>
public class DeleteMultipleObjects
{
    /// <summary>
    /// The Main method initializes the Amazon S3 client and the name of
    /// the bucket and then passes those values to MultiObjectDeleteAsync.
    /// </summary>
    public static async Task Main()
    {
        const string bucketName = "amzn-s3-demo-bucket";

        // If the Amazon S3 bucket from which you wish to delete objects is
not
        // located in the same AWS Region as the default user, define the
        // AWS Region for the Amazon S3 bucket as a parameter to the client
        // constructor.

```

```
        IAmazonS3 s3Client = new AmazonS3Client();

        await MultiObjectDeleteAsync(s3Client, bucketName);
    }

    /// <summary>
    /// This method uses the passed Amazon S3 client to first create and then
    /// delete three files from the named bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// Amazon S3 methods.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket where
objects
    /// will be created and then deleted.</param>
    public static async Task MultiObjectDeleteAsync(IAmazonS3 client, string
bucketName)
    {
        // Create three sample objects which we will then delete.
        var keysAndVersions = await PutObjectsAsync(client, 3, bucketName);

        // Now perform the multi-object delete, passing the key names and
        // version IDs. Since we are working with a non-versioned bucket,
        // the object keys collection includes null version IDs.
        DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest
        {
            BucketName = bucketName,
            Objects = keysAndVersions,
        };

        // You can add a specific object key to the delete request using the
        // AddKey method of the multiObjectDeleteRequest.
        try
        {
            DeleteObjectsResponse response = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
            Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
        }
        catch (DeleteObjectsException e)
        {
            PrintDeletionErrorStatus(e);
        }
    }
}
```

```
    }

    /// <summary>
    /// Prints the list of errors raised by the call to DeleteObjectsAsync.
    /// </summary>
    /// <param name="ex">A collection of exceptions returned by the call to
    /// DeleteObjectsAsync.</param>
    public static void PrintDeletionErrorStatus(DeleteObjectsException ex)
    {
        DeleteObjectsResponse errorResponse = ex.Response;
        Console.WriteLine("x {0}", errorResponse.DeletedObjects.Count);

        Console.WriteLine($"Successfully deleted
{errorResponse.DeletedObjects.Count}.");
        Console.WriteLine($"No. of objects failed to delete =
{errorResponse.DeleteErrors.Count}");

        Console.WriteLine("Printing error data...");
        foreach (DeleteError deleteError in errorResponse.DeleteErrors)
        {
            Console.WriteLine($"Object Key:
{deleteError.Key}\t{deleteError.Code}\t{deleteError.Message}");
        }
    }

    /// <summary>
    /// This method creates simple text file objects that can be used in
    /// the delete method.
    /// </summary>
    /// <param name="client">The Amazon S3 client used to call
    PutObjectAsync.</param>
    /// <param name="number">The number of objects to create.</param>
    /// <param name="bucketName">The name of the bucket where the objects
    /// will be created.</param>
    /// <returns>A list of keys (object keys) and versions that the calling
    /// method will use to delete the newly created files.</returns>
    public static async Task<List<KeyVersion>> PutObjectsAsync(IAmazonS3
    client, int number, string bucketName)
    {
        List<KeyVersion> keys = new List<KeyVersion>();
        for (int i = 0; i < number; i++)
        {
            string key = "ExampleObject-" + new System.Random().Next();
            PutObjectRequest request = new PutObjectRequest
```

```

        {
            BucketName = bucketName,
            Key = key,
            ContentBody = "This is the content body!",
        };

        PutObjectResponse response = await
client.PutObjectAsync(request);

        // For non-versioned bucket operations, we only need the
        // object key.
        KeyVersion keyVersion = new KeyVersion
        {
            Key = key,
        };
        keys.Add(keyVersion);
    }

    return keys;
}
}

```

Löschen Sie mehrere Objekte in einem versionierten S3-Bucket.

```

using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to delete objects in a version-enabled Amazon
/// Simple StorageService (Amazon S3) bucket.
/// </summary>
public class DeleteMultipleObjects
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";

        // If the AWS Region for your Amazon S3 bucket is different from

```

```
// the AWS Region of the default user, define the AWS Region for
// the Amazon S3 bucket and pass it to the client constructor
// like this:
// RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
IAmazonS3 s3Client;

s3Client = new AmazonS3Client();
await DeleteMultipleObjectsFromVersionedBucketAsync(s3Client,
bucketName);
}

/// <summary>
/// This method removes multiple versions and objects from a
/// version-enabled Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used to
call
/// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
/// RemoveDeleteMarkersAsync.</param>
/// <param name="bucketName">The name of the bucket from which to delete
/// objects.</param>
public static async Task
DeleteMultipleObjectsFromVersionedBucketAsync(IAmazonS3 client, string
bucketName)
{
    // Delete objects (specifying object version in the request).
    await DeleteObjectVersionsAsync(client, bucketName);

    // Delete objects (without specifying object version in the request).
    var deletedObjects = await DeleteObjectsAsync(client, bucketName);

    // Additional exercise - remove the delete markers Amazon S3 returned
from
    // the preceding response. This results in the objects reappearing
    // in the bucket (you can verify the appearance/disappearance of
    // objects in the console).
    await RemoveDeleteMarkersAsync(client, bucketName, deletedObjects);
}

/// <summary>
/// Creates and then deletes non-versioned Amazon S3 objects and then
deletes
/// them again. The method returns a list of the Amazon S3 objects
deleted.
```

```

    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// PubObjectsAsync and NonVersionedDeleteAsync.</param>
    /// <param name="bucketName">The name of the bucket where the objects
    /// will be created and then deleted.</param>
    /// <returns>A list of DeletedObjects.</returns>
    public static async Task<List<DeletedObject>>
DeleteObjectsAsync(IAmazonS3 client, string bucketName)
    {
        // Upload the sample objects.
        var keysAndVersions2 = await PutObjectsAsync(client, bucketName, 3);

        // Delete objects using only keys. Amazon S3 creates a delete marker
and
        // returns its version ID in the response.
        List<DeletedObject> deletedObjects = await
NonVersionedDeleteAsync(client, bucketName, keysAndVersions2);
        return deletedObjects;
    }

    /// <summary>
    /// This method creates several temporary objects and then deletes them.
    /// </summary>
    /// <param name="client">The S3 client.</param>
    /// <param name="bucketName">Name of the bucket.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteObjectVersionsAsync(IAmazonS3 client,
string bucketName)
    {
        // Upload the sample objects.
        var keysAndVersions1 = await PutObjectsAsync(client, bucketName, 3);

        // Delete the specific object versions.
        await VersionedDeleteAsync(client, bucketName, keysAndVersions1);
    }

    /// <summary>
    /// Displays the list of information about deleted files to the console.
    /// </summary>
    /// <param name="e">Error information from the delete process.</param>
    private static void DisplayDeletionErrors(DeleteObjectsException e)
    {
        var errorResponse = e.Response;

```

```

        Console.WriteLine($"No. of objects successfully deleted =
{errorResponse.DeletedObjects.Count}");
        Console.WriteLine($"No. of objects failed to delete =
{errorResponse.DeleteErrors.Count}");
        Console.WriteLine("Printing error data...");
        foreach (var deleteError in errorResponse.DeleteErrors)
        {
            Console.WriteLine($"Object Key:
{deleteError.Key}\\t{deleteError.Code}\\t{deleteError.Message}");
        }
    }

    /// <summary>
    /// Delete multiple objects from a version-enabled bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
    /// RemoveDeleteMarkersAsync.</param>
    /// <param name="bucketName">The name of the bucket from which to delete
    /// objects.</param>
    /// <param name="keys">A list of key names for the objects to delete.</
param>
    private static async Task VersionedDeleteAsync(IAmazonS3 client, string
bucketName, List<KeyVersion> keys)
    {
        var multiObjectDeleteRequest = new DeleteObjectsRequest
        {
            BucketName = bucketName,
            Objects = keys, // This includes the object keys and specific
version IDs.
        };

        try
        {
            Console.WriteLine("Executing VersionedDelete...");
            DeleteObjectsResponse response = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
            Console.WriteLine($"Successfully deleted all the
{response.DeletedObjects.Count} items");
        }
        catch (DeleteObjectsException ex)
        {
            DisplayDeletionErrors(ex);
        }
    }

```



```

    }
}

/// <summary>
/// Deletes multiple objects from a non-versioned Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used to
call
/// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
/// RemoveDeleteMarkersAsync.</param>
/// <param name="bucketName">The name of the bucket from which to delete
/// objects.</param>
/// <param name="keys">A list of key names for the objects to delete.</
param>
/// <returns>A list of the deleted objects.</returns>
private static async Task<List<DeletedObject>>
NonVersionedDeleteAsync(IAmazonS3 client, string bucketName, List<KeyVersion>
keys)
{
    // Create a request that includes only the object key names.
    DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest();
    multiObjectDeleteRequest.BucketName = bucketName;

    foreach (var key in keys)
    {
        multiObjectDeleteRequest.AddKey(key.Key);
    }

    // Execute DeleteObjectsAsync.
    // The DeleteObjectsAsync method adds a delete marker for each
    // object deleted. You can verify that the objects were removed
    // using the Amazon S3 console.
    DeleteObjectsResponse response;
    try
    {
        Console.WriteLine("Executing NonVersionedDelete...");
        response = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
        Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
    }
    catch (DeleteObjectsException ex)
    {

```

```
        DisplayDeletionErrors(ex);
        throw; // Some deletions failed. Investigate before continuing.
    }

    // This response contains the DeletedObjects list which we use to
delete the delete markers.
    return response.DeletedObjects;
}

/// <summary>
/// Deletes the markers left after deleting the temporary objects.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used to
call
/// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
/// RemoveDeleteMarkersAsync.</param>
/// <param name="bucketName">The name of the bucket from which to delete
/// objects.</param>
/// <param name="deletedObjects">A list of the objects that were
deleted.</param>
private static async Task RemoveDeleteMarkersAsync(IAmazonS3 client,
string bucketName, List<DeletedObject> deletedObjects)
{
    var keyVersionList = new List<KeyVersion>();

    foreach (var deletedObject in deletedObjects)
    {
        KeyVersion keyVersion = new KeyVersion
        {
            Key = deletedObject.Key,
            VersionId = deletedObject.DeleteMarkerVersionId,
        };
        keyVersionList.Add(keyVersion);
    }

    // Create another request to delete the delete markers.
    var multiObjectDeleteRequest = new DeleteObjectsRequest
    {
        BucketName = bucketName,
        Objects = keyVersionList,
    };

    // Now, delete the delete marker to bring your objects back to the
bucket.
```

```

        try
        {
            Console.WriteLine("Removing the delete markers .....");
            var deleteObjectResponse = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
            Console.WriteLine($"Successfully deleted the
{deleteObjectResponse.DeletedObjects.Count} delete markers");
        }
        catch (DeleteObjectsException ex)
        {
            DisplayDeletionErrors(ex);
        }
    }

    /// <summary>
    /// Create temporary Amazon S3 objects to show how object deletion works
in an
    /// Amazon S3 bucket with versioning enabled.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// PutObjectAsync to create temporary objects for the example.</param>
    /// <param name="bucketName">A string representing the name of the S3
    /// bucket where we will create the temporary objects.</param>
    /// <param name="number">The number of temporary objects to create.</
param>
    /// <returns>A list of the KeyVersion objects.</returns>
    private static async Task<List<KeyVersion>> PutObjectsAsync(IAmazonS3
client, string bucketName, int number)
    {
        var keys = new List<KeyVersion>();

        for (var i = 0; i < number; i++)
        {
            string key = "ObjectToDelete-" + new System.Random().Next();
            PutObjectRequest request = new PutObjectRequest
            {
                BucketName = bucketName,
                Key = key,
                ContentBody = "This is the content body!",
            };

            var response = await client.PutObjectAsync(request);
            KeyVersion keyVersion = new KeyVersion

```

```

        {
            Key = key,
            VersionId = response.VersionId,
        };

        keys.Add(keyVersion);
    }

    return keys;
}
}

```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.

```

```
#      $2 - A list of keys in the bucket to delete.

# Returns:
#      0 - If successful.
#      1 - If it fails.
#####
function delete_items_in_bucket() {
    local bucket_name=$1
    local keys=$2
    local response

    # Create the JSON for the items to delete.
    local delete_items
    delete_items="{\"Objects\":["
    for key in $keys; do
        delete_items="$delete_items{\"Key\": \"$key\"},"
    done
    delete_items=${delete_items%?} # Remove the final comma.
    delete_items="$delete_items]"

    response=$(aws s3api delete-objects \
        --bucket "$bucket_name" \
        --delete "$delete_items")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n
$response"
        return 1
    fi
}
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS CLI Befehlsreferenz.

C++

SDK für C++

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::deleteObjects(const std::vector<Aws::String> &objectKeys,
                               const Aws::String &fromBucket,
                               const Aws::S3::S3ClientConfiguration
                               &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::DeleteObjectsRequest request;

    Aws::S3::Model::Delete deleteObject;
    for (const Aws::String &objectKey: objectKeys) {
        deleteObject.AddObjects(Aws::S3::Model::ObjectIdentifier().WithKey(objectKey));
    }

    request.SetDelete(deleteObject);
    request.SetBucket(fromBucket);

    Aws::S3::Model::DeleteObjectsOutcome outcome =
        client.DeleteObjects(request);

    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error deleting objects. " <<
            err.GetExceptionName() << ": " << err.GetMessage() <<
        std::endl;
    } else {
        std::cout << "Successfully deleted the objects.";
        for (size_t i = 0; i < objectKeys.size(); ++i) {
            std::cout << objectKeys[i];
            if (i < objectKeys.size() - 1) {
                std::cout << ", ";
            }
        }
    }
}
```

```
        std::cout << " from bucket " << fromBucket << "." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl löscht ein Objekt aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json` ist ein JSON-Dokument im aktuellen Verzeichnis, das das zu löschende Objekt angibt:

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```


Ausgabe:

```
{
  "Deleted": [
    {
      "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUP01koHwzU",
      "Key": "test1.txt",
      "DeleteMarker": true
    }
  ]
}
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS CLI Befehlsreferenz.

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
    S3Client    *s3.Client
    S3Manager   *manager.Uploader
}

// DeleteObjects deletes a list of objects from a bucket.
func (actor S3Actions) DeleteObjects(ctx context.Context, bucket string, objects
[]types.ObjectIdentifier, bypassGovernance bool) error {
    if len(objects) == 0 {
        return nil
    }

    input := s3.DeleteObjectsInput{
        Bucket: aws.String(bucket),
        Delete: &types.Delete{
            Objects: objects,
            Quiet:   aws.Bool(true),
        },
    }
    if bypassGovernance {
        input.BypassGovernanceRetention = aws.Bool(true)
    }
    delOut, err := actor.S3Client.DeleteObjects(ctx, &input)
    if err != nil || len(delOut.Errors) > 0 {
        log.Printf("Error deleting objects from bucket %s.\n", bucket)
        if err != nil {
```



```
var noBucket *types.NoSuchBucket
if errors.As(err, &noBucket) {
    log.Printf("Bucket %s does not exist.\n", bucket)
    err = noBucket
}
} else if len(delOut.Errors) > 0 {
    for _, outErr := range delOut.Errors {
        log.Printf("%s: %s\n", *outErr.Key, *outErr.Message)
    }
    err = fmt.Errorf("%s", *delOut.Errors[0].Message)
}
}
return err
}
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.services.s3.model.Delete;
import software.amazon.awssdk.services.s3.model.DeleteObjectsRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.ArrayList;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/

public class DeleteMultiObjects {
    public static void main(String[] args) {
        final String usage = ""

            Usage:    <bucketName>

            Where:
                bucketName - the Amazon S3 bucket name.
            "";

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        deleteBucketObjects(s3, bucketName);
        s3.close();
    }

    public static void deleteBucketObjects(S3Client s3, String bucketName) {
        // Upload three sample objects to the specified Amazon S3 bucket.
        ArrayList<ObjectIdentifier> keys = new ArrayList<>();
        PutObjectRequest putObj;
        ObjectIdentifier objectId;

        for (int i = 0; i < 3; i++) {
            String keyName = "delete object example " + i;
            objectId = ObjectIdentifier.builder()
                .key(keyName)
                .build();
        }
    }
}
```

```
        putOb = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(keyName)
            .build();

        s3.putObject(putOb, RequestBody.fromString(keyName));
        keys.add(objectId);
    }

    System.out.println(keys.size() + " objects successfully created.");

    // Delete multiple objects in one request.
    Delete del = Delete.builder()
        .objects(keys)
        .build();

    try {
        DeleteObjectsRequest multiObjectDeleteRequest =
DeleteObjectsRequest.builder()
            .bucket(bucketName)
            .delete(del)
            .build();

        s3.deleteObjects(multiObjectDeleteRequest);
        System.out.println("Multiple objects are deleted!");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie mehrere Objekte.

```
import { DeleteObjectsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new DeleteObjectsCommand({
    Bucket: "test-bucket",
    Delete: {
      Objects: [{ Key: "object1.txt" }, { Key: "object2.txt" }],
    },
  });

  try {
    const { Deleted } = await client.send(command);
    console.log(
      `Successfully deleted ${Deleted.length} objects from S3 bucket. Deleted objects:`,
    );
    console.log(Deleted.map((d) => ` • ${d.Key}`).join("\n"));
  } catch (err) {
    console.error(err);
  }
};
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun deleteBucketObjects(
    bucketName: String,
    objectName: String,
) {
    val objectId =
        ObjectIdentifier {
            key = objectName
        }

    val delOb =
        Delete {
            objects = listOf(objectId)
        }


    val request =
        DeleteObjectsRequest {
            bucket = bucketName
            delete = delOb
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteObjects(request)
        println("$objectName was deleted from $bucketName")
    }
}
```

- API Einheiten finden Sie [DeleteObjects](#) unter AWS SDK als API Kotlin-Referenz.

PHP

SDK für PHP

 Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie eine Reihe von Objekten aus einer Schlüsselliste.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $objects = [];
    foreach ($contents['Contents'] as $content) {
        $objects[] = [
            'Key' => $content['Key'],
        ];
    }
    $this->s3client->deleteObjects([
        'Bucket' => $this->bucketName,
        'Delete' => [
            'Objects' => $objects,
        ],
    ]);
    $check = $this->s3client->listObjectsV2([
        'Bucket' => $this->bucketName,
    ]);
    if (count($check) <= 0) {
        throw new Exception("Bucket wasn't empty.");
    }
    echo "Deleted all objects and folders from $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to delete $fileName from $this->bucketName with error:
" . $exception->getMessage();
    exit("Please fix error with object deletion before continuing.");
}
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl entfernt das Objekt "sample.txt" aus dem Bucket „test-files“. Sie werden zur Bestätigung aufgefordert, bevor der Befehl ausgeführt wird. Um die Aufforderung zu unterdrücken, verwenden Sie den Schalter -Force.

```
Remove-S3Object -BucketName test-files -Key sample.txt
```

Beispiel 2: Dieser Befehl entfernt die angegebene Version des Objekts "sample.txt" aus dem Bucket „test-files“, vorausgesetzt, der Bucket wurde so konfiguriert, dass Objektversionen aktiviert werden.

```
Remove-S3Object -BucketName test-files -Key sample.txt -VersionId  
HLbxnx6V9omT6AQYVpks8mmFKQcejpqt
```

Beispiel 3: Dieser Befehl entfernt die Objekte "sample1.txt", "sample2.txt" und "sample3.txt" aus dem Bucket „test-files“ als einzelne Batch-Operation. In der Serviceantwort werden alle verarbeiteten Schlüssel aufgeführt, unabhängig vom Erfolgs- oder Fehlerstatus des Löschvorgangs. Um nur Fehler für Schlüssel zu erhalten, die vom Dienst nicht verarbeitet werden konnten, fügen Sie den ReportErrorsOnly Parameter hinzu (dieser Parameter kann auch mit dem Alias -Quiet angegeben werden).

```
Remove-S3Object -BucketName test-files -KeyCollection @( "sample1.txt",  
"sample2.txt", "sample3.txt" )
```

Beispiel 4: In diesem Beispiel wird ein Inline-Ausdruck mit dem KeyCollection Parameter verwendet, um die Schlüssel der zu löschenden Objekte abzurufen. Get-S3Object gibt eine Sammlung von Amazon.S3.Model.S3Object-Instanzen zurück, von denen jede ein Key-Element vom Typ Zeichenfolge hat, das das Objekt identifiziert.

```
Remove-S3Object -bucketname "test-files" -KeyCollection (Get-S3Object "test-  
files" -KeyPrefix "prefix/subprefix" | select -ExpandProperty Key)
```

Beispiel 5: In diesem Beispiel werden alle Objekte abgerufen, die ein key prefix „Präfix/ Unterpräfix“ im Bucket haben, und sie werden gelöscht. Beachten Sie, dass die eingehenden Objekte nacheinander verarbeitet werden. Bei großen Sammlungen sollten Sie erwägen, die Sammlung an den Parameter - InputObject (alias -S3ObjectCollection) des Cmdlets zu

übergeben, damit das Löschen als Batch mit einem einzigen Aufruf des Dienstes erfolgen kann.

```
Get-S3Object -BucketName "test-files" -KeyPrefix "prefix/subprefix" | Remove-S3Object -Force
```

Beispiel 6: In diesem Beispiel wird eine Sammlung von ObjectVersion Amazon.S3.Model.S3-Instances, die Löschmarkierungen darstellen, zur Löschung an das Cmdlet übergeben. Beachten Sie, dass die eingehenden Objekte nacheinander verarbeitet werden. Bei großen Sammlungen sollten Sie erwägen, die Sammlung an den Parameter - InputObject (alias - S3ObjectCollection) des Cmdlets zu übergeben, damit das Löschen als Batch mit einem einzigen Aufruf des Dienstes erfolgen kann.

```
(Get-S3Version -BucketName "test-files").Versions | Where {$_.IsDeleteMarker -eq "True"} | Remove-S3Object -Force
```

Beispiel 7: Dieses Skript zeigt, wie eine Gruppe von Objekten (in diesem Fall Löschmarken) im Batch-Modus gelöscht werden kann, indem ein Array von Objekten erstellt wird, die mit dem Parameter - verwendet werden sollen. KeyAndVersionCollection

```
$keyVersions = @()
$markers = (Get-S3Version -BucketName $BucketName).Versions | Where
  {$_.IsDeleteMarker -eq "True"}
foreach ($marker in $markers) { $keyVersions += @{ Key = $marker.Key; VersionId =
  $marker.VersionId } }
Remove-S3Object -BucketName $BucketName -KeyAndVersionCollection $keyVersions -
Force
```

- API-Einheiten finden Sie in [DeleteObjects](#) AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löschen Sie eine Reihe von Objekten mithilfe einer Liste von Objektschlüsseln.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
            that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    @staticmethod
    def delete_objects(bucket, object_keys):
        """
        Removes a list of objects from a bucket.
        This operation is done as a batch in a single request.

        :param bucket: The bucket that contains the objects. This is a Boto3
        Bucket
            resource.
        :param object_keys: The list of keys that identify the objects to remove.
        :return: The response that contains data about which objects were deleted
        and any that could not be deleted.
        """
        try:
            response = bucket.delete_objects(
                Delete={"Objects": [{"Key": key} for key in object_keys]}
            )
            if "Deleted" in response:
                logger.info(
                    "Deleted objects '%s' from bucket '%s'.",
                    [del_obj["Key"] for del_obj in response["Deleted"]],
                    bucket.name,
                )
            if "Errors" in response:
                logger.warning(
                    "Could not delete objects '%s' from bucket '%s'.",
                    [
                        f"{del_obj['Key']}: {del_obj['Code']}"
                        for del_obj in response["Errors"]
                    ]
                )
```

```

        ],
        bucket.name,
    )
except ClientError:
    logger.exception("Couldn't delete any objects from bucket %s.",
bucket.name)
    raise
else:
    return response

```

Löschen Sie alle Objekte im Bucket.

```

class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
in Boto3
                                that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    @staticmethod
    def empty_bucket(bucket):
        """
        Remove all objects from a bucket.

        :param bucket: The bucket to empty. This is a Boto3 Bucket resource.
        """
        try:
            bucket.objects.delete()
            logger.info("Emptied bucket '%s'.", bucket.name)
        except ClientError:
            logger.exception("Couldn't empty bucket '%s'.", bucket.name)
            raise

```

Löschen Sie ein versioniertes Objekt dauerhaft, indem Sie alle seine Versionen löschen.

```
def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?")
  answer = gets.chomp.downcase
```

```

    if answer == "y"
      bucket.objects.batch_delete!
      bucket.delete
      puts("Emptied and deleted bucket #{bucket.name}.\n")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't empty and delete bucket #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
end

```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

/// Delete the objects in a bucket.
pub async fn delete_objects(
    client: &aws_sdk_s3::Client,
    bucket_name: &str,
    objects_to_delete: Vec<String>,
) -> Result<(), S3ExampleError> {
    // Push into a mut vector to use `?` early return errors while building
    object keys.
    let mut delete_object_ids: Vec<aws_sdk_s3::types::ObjectIdentifier> = vec![];
    for obj in objects_to_delete {
        let obj_id = aws_sdk_s3::types::ObjectIdentifier::builder()
            .key(obj)
            .build()
            .map_err(|err| {
                S3ExampleError::new(format!("Failed to build key for
delete_object: {err:?}"))
            })?;
        delete_object_ids.push(obj_id);
    }
}

```

```

    }

    client
      .delete_objects()
      .bucket(bucket_name)
      .delete(
        aws_sdk_s3::types::Delete::builder()
          .set_objects(Some(delete_object_ids))
          .build()
          .map_err(|err| {
            S3ExampleError::new(format!("Failed to build delete_object
input {err:?}"))
          })?,
      )
      .send()
      .await?;
    Ok(())
  }

```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK als API Referenz auf Rust.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

import AWSS3

public func deleteObjects(bucket: String, keys: [String]) async throws {
    let input = DeleteObjectsInput(
        bucket: bucket,
        delete: S3ClientTypes.Delete(
            objects: keys.map { S3ClientTypes.ObjectIdentifier(key: $0) },
            quiet: true
        )
    )
}

```

```
do {
    _ = try await client.deleteObjects(input: input)
} catch {
    print("ERROR: deleteObjects:", dump(error))
    throw error
}
```

- API-Einheiten finden Sie [DeleteObjects](#) unter AWS SDK als API-Swift-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeletePublicAccessBlock** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `DeletePublicAccessBlock`.

CLI

AWS CLI

Um die Konfiguration „Öffentlichen Zugriff blockieren“ für einen Bucket zu löschen

Im folgenden `delete-public-access-block` Beispiel wird die Konfiguration „Öffentlicher Zugriff blockieren“ für den angegebenen Bucket entfernt.

```
aws s3api delete-public-access-block \
  --bucket my-bucket
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePublicAccessBlock](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl deaktiviert die Einstellung „Öffentlichen Zugriff blockieren“ für den angegebenen Bucket.

```
Remove-S3PublicAccessBlock -BucketName 's3testbucket' -Force -Select  
'^BucketName'
```

Ausgabe:

```
s3testbucket
```

- API-Einheiten finden Sie [DeletePublicAccessBlock](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketAccelerateConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketAccelerateConfiguration`.

CLI

AWS CLI

Um die Accelerate-Konfiguration eines Buckets abzurufen

Im folgenden `get-bucket-accelerate-configuration` Beispiel wird die Accelerate-Konfiguration für den angegebenen Bucket abgerufen.

```
aws s3api get-bucket-accelerate-configuration \  
--bucket my-bucket
```

Ausgabe:

```
{  
  "Status": "Enabled"  
}
```

- API-Einheiten finden Sie unter [GetBucketAccelerateConfiguration](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt den Wert Enabled zurück, wenn die Einstellungen für die Übertragungsbeschleunigung für den angegebenen Bucket aktiviert sind.

```
Get-S3BucketAccelerateConfiguration -BucketName 's3testbucket'
```

Ausgabe:

```
Value  
-----  
Enabled
```

- API-Einheiten finden Sie [GetBucketAccelerateConfiguration](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketAc1** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketAc1`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Zugriffskontrolllisten verwalten \(\) ACLs](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).


```

    /// <summary>
    /// Get the access control list (ACL) for the new bucket.
    /// </summary>
    /// <param name="client">The initialized client object used to get the
    /// access control list (ACL) of the bucket.</param>
    /// <param name="newBucketName">The name of the newly created bucket.</
param>
    /// <returns>An S3AccessControlList.</returns>
    public static async Task<S3AccessControlList>
    GetACLForBucketAsync(IAmazonS3 client, string newBucketName)
    {
        // Retrieve bucket ACL to show that the ACL was properly applied to
        // the new bucket.
        GetACLResponse getACLResponse = await client.GetACLAsync(new
    GetACLRequest
    {
        BucketName = newBucketName,
    });

        return getACLResponse.AccessControlList;
    }

```

- API-Einheiten finden Sie [GetBucketAcl](#) unter AWS SDK for .NET API-Referenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

bool AwsDoc::S3::getBucketAcl(const Aws::String &bucketName,
                             const Aws::S3::S3ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client s3Client(clientConfig);

```

```
Aws::S3::Model::GetBucketAclRequest request;
request.SetBucket(bucketName);

Aws::S3::Model::GetBucketAclOutcome outcome =
    s3Client.GetBucketAcl(request);

if (!outcome.IsSuccess()) {
    const Aws::S3::S3Error &err = outcome.GetError();
    std::cerr << "Error: getBucketAcl: "
                << err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
} else {
    Aws::Vector<Aws::S3::Model::Grant> grants =
        outcome.GetResult().GetGrants();

    for (auto it = grants.begin(); it != grants.end(); it++) {
        Aws::S3::Model::Grant grant = *it;
        Aws::S3::Model::Grantee grantee = grant.GetGrantee();

        std::cout << "For bucket " << bucketName << ": "
                  << std::endl << std::endl;

        if (grantee.TypeHasBeenSet()) {
            std::cout << "Type:          "
                      << getGranteeTypeString(grantee.GetType()) <<
std::endl;
        }

        if (grantee.DisplayNameHasBeenSet()) {
            std::cout << "Display name: "
                      << grantee.GetDisplayName() << std::endl;
        }

        if (grantee.EmailAddressHasBeenSet()) {
            std::cout << "Email address: "
                      << grantee.GetEmailAddress() << std::endl;
        }

        if (grantee.IDHasBeenSet()) {
            std::cout << "ID:          "
                      << grantee.GetID() << std::endl;
        }
    }
}
```

```
        if (grantee.URIHasBeenSet()) {
            std::cout << "URI:          "
                << grantee.GetURI() << std::endl;
        }

        std::cout << "Permission:    " <<
            getPermissionString(grant.GetPermission()) <<
            std::endl << std::endl;
    }
}

return outcome.IsSuccess();
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \param type: Type enumeration.
 \return String: Human-readable string.
 */

Aws::String getGranteeTypeString(const Aws::S3::Model::Type &type) {
    switch (type) {
        case Aws::S3::Model::Type::AmazonCustomerByEmail:
            return "Email address of an AWS account";
        case Aws::S3::Model::Type::CanonicalUser:
            return "Canonical user ID of an AWS account";
        case Aws::S3::Model::Type::Group:
            return "Predefined Amazon S3 group";
        case Aws::S3::Model::Type::NOT_SET:
            return "Not set";
        default:
            return "Type unknown";
    }
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \param permission: Permission enumeration.
 \return String: Human-readable string.
 */

Aws::String getPermissionString(const Aws::S3::Model::Permission &permission) {
```

```
switch (permission) {
    case Aws::S3::Model::Permission::FULL_CONTROL:
        return "Can list objects in this bucket, create/overwrite/delete "
            "objects in this bucket, and read/write this "
            "bucket's permissions";
    case Aws::S3::Model::Permission::NOT_SET:
        return "Permission not set";
    case Aws::S3::Model::Permission::READ:
        return "Can list objects in this bucket";
    case Aws::S3::Model::Permission::READ_ACP:
        return "Can read this bucket's permissions";
    case Aws::S3::Model::Permission::WRITE:
        return "Can create, overwrite, and delete objects in this bucket";
    case Aws::S3::Model::Permission::WRITE_ACP:
        return "Can write this bucket's permissions";
    default:
        return "Permission unknown";
}

return "Permission unknown";
}
```

- API-Einheiten finden Sie [GetBucketAcl](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl ruft die Zugriffskontrollliste für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-acl --bucket my-bucket
```

Ausgabe:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
}
```

```
"Grants": [  
  {  
    "Grantee": {  
      "DisplayName": "my-username",  
      "ID":  
"7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"  
    },  
    "Permission": "FULL_CONTROL"  
  }  
]  
}
```

- API-Einheiten finden Sie unter [GetBucketAcl](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.services.s3.model.S3Exception;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3.S3Client;  
import software.amazon.awssdk.services.s3.model.GetObjectAclRequest;  
import software.amazon.awssdk.services.s3.model.GetObjectAclResponse;  
import software.amazon.awssdk.services.s3.model.Grant;  
import java.util.List;  
  
/**  
 * Before running this Java V2 code example, set up your development  
 * environment, including your credentials.  
 *  
 * For more information, see the following documentation topic:  
 *  
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html  
 */
```

```
public class GetAcl {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <objectKey>

            Where:
                bucketName - The Amazon S3 bucket to get the access control
list (ACL) for.
                objectKey - The object to get the ACL for.\s
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String objectKey = args[1];
        System.out.println("Retrieving ACL for object: " + objectKey);
        System.out.println("in bucket: " + bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        getBucketACL(s3, objectKey, bucketName);
        s3.close();
        System.out.println("Done!");
    }

    public static String getBucketACL(S3Client s3, String objectKey, String
bucketName) {
        try {
            GetObjectAclRequest aclReq = GetObjectAclRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();

            GetObjectAclResponse aclRes = s3.getObjectAcl(aclReq);
            List<Grant> grants = aclRes.grants();
            String grantee = "";
            for (Grant grant : grants) {
```

```
        System.out.format("  %s: %s\n", grant.grantee().id(),
grant.permission());
        grantee = grant.grantee().id();
    }

    return grantee;
} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}

return "";
}
}
```

- API-Einheiten finden Sie [GetBucketAcl](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Holen Sie sich die ACL-Berechtigungen.

```
import { GetBucketAclCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
    const command = new GetBucketAclCommand({
        Bucket: "test-bucket",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    }
}
```

```
    } catch (err) {  
        console.error(err);  
    }  
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [GetBucketAcl](#) in AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:  
    """Encapsulates S3 bucket actions."""  
  
    def __init__(self, bucket):  
        """  
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in  
Boto3  
                that wraps bucket actions in a class-like structure.  
        """  
        self.bucket = bucket  
        self.name = bucket.name  
  
    def get_acl(self):  
        """  
        Get the ACL of the bucket.  
  
        :return: The ACL of the bucket.  
        """  
        try:  
            acl = self.bucket.Acl()  
            logger.info(  

```



```
        "Got ACL for bucket %s. Owner is %s.", self.bucket.name,
acl.owner
    )
    except ClientError:
        logger.exception("Couldn't get ACL for bucket %s.", self.bucket.name)
        raise
    else:
        return acl
```

- API-Einheiten finden Sie [GetBucketAcl](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketAnalyticsConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketAnalyticsConfiguration`.

CLI

AWS CLI

Um die Analytics-Konfiguration für einen Bucket mit einer bestimmten ID abzurufen

Im folgenden `get-bucket-analytics-configuration` Beispiel wird die Analytics-Konfiguration für den angegebenen Bucket und die angegebene ID angezeigt.

```
aws s3api get-bucket-analytics-configuration \
  --bucket my-bucket \
  --id 1
```

Ausgabe:

```
{
  "AnalyticsConfiguration": {
    "StorageClassAnalysis": {},
  },
}
```

```
    "Id": "1"  
  }  
}
```

- API-Einheiten finden Sie [GetBucketAnalyticsConfiguration](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die Details des Analysefilters mit dem Namen 'testfilter' im angegebenen S3-Bucket zurück.

```
Get-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId  
'testfilter'
```

- API-Einheiten finden Sie unter [GetBucketAnalyticsConfiguration](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketCors** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketCors`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
```

```
/// Retrieve the CORS configuration applied to the Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used
/// to retrieve the CORS configuration.</param>
/// <returns>The created CORS configuration object.</returns>
private static async Task<CORSConfiguration>
RetrieveCORSConfigurationAsync(AmazonS3Client client)
{
    GetCORSConfigurationRequest request = new
GetCORSConfigurationRequest()
    {
        BucketName = BucketName,
    };
    var response = await client.GetCORSConfigurationAsync(request);
    var configuration = response.Configuration;
    PrintCORSRules(configuration);
    return configuration;
}
```

- API-Einheiten finden Sie [GetBucketCors](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Der folgende Befehl ruft die Cross-Origin Resource Sharing-Konfiguration für einen Bucket mit dem Namen ab: `my-bucket`

```
aws s3api get-bucket-cors --bucket my-bucket
```

Ausgabe:

```
{
  "CORSRules": [
    {
      "AllowedHeaders": [
        "*"
      ],
      "ExposeHeaders": [
        "x-amz-server-side-encryption"
      ]
    }
  ]
}
```

```
    ],
    "AllowedMethods": [
      "PUT",
      "POST",
      "DELETE"
    ],
    "MaxAgeSeconds": 3000,
    "AllowedOrigins": [
      "http://www.example.com"
    ]
  },
  {
    "AllowedHeaders": [
      "Authorization"
    ],
    "MaxAgeSeconds": 3000,
    "AllowedMethods": [
      "GET"
    ],
    "AllowedOrigins": [
      "*"
    ]
  }
]
```

- API-Einheiten finden Sie unter [GetBucketCors](#) AWS CLI Befehlsreferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Holen Sie sich die CORS Richtlinie für den Bucket.

```
import { GetBucketCorsCommand, S3Client } from "@aws-sdk/client-s3";
```

```
const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketCorsCommand({
    Bucket: "test-bucket",
  });

  try {
    const { CORSRules } = await client.send(command);
    CORSRules.forEach((cr, i) => {
      console.log(
        `\nCORSRule ${i + 1}`,
        `\n${"-".repeat(10)}`,
        `\nAllowedHeaders: ${cr.AllowedHeaders.join(" ")}`,
        `\nAllowedMethods: ${cr.AllowedMethods.join(" ")}`,
        `\nAllowedOrigins: ${cr.AllowedOrigins.join(" ")}`,
        `\nExposeHeaders: ${cr.ExposeHeaders.join(" ")}`,
        `\nMaxAgeSeconds: ${cr.MaxAgeSeconds}`,
      );
    });
  } catch (err) {
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [GetBucketCors](#) in AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
```

```
def __init__(self, bucket):
    """
    :param bucket: A Boto3 Bucket resource. This is a high-level resource in
    Boto3
                   that wraps bucket actions in a class-like structure.
    """
    self.bucket = bucket
    self.name = bucket.name

def get_cors(self):
    """
    Get the CORS rules for the bucket.

    :return The CORS rules for the specified bucket.
    """
    try:
        cors = self.bucket.Cors()
        logger.info(
            "Got CORS rules %s for bucket '%s'.", cors.cors_rules,
            self.bucket.name
        )
    except ClientError:
        logger.exception(("Couldn't get CORS for bucket %s.",
            self.bucket.name))
        raise
    else:
        return cors
```

- API-Einheiten finden Sie [GetBucketCors](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
  an existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Gets the CORS configuration of a bucket.
  #
  # @return [Aws::S3::Type::GetBucketCorsOutput, nil] The current CORS
  configuration for the bucket.
  def get_cors
    @bucket_cors.data
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get CORS configuration for #{@bucket_cors.bucket.name}. Here's
  why: #{e.message}"
    nil
  end
end

end
```

- API Einzelheiten finden Sie [GetBucketCors](#) unter AWS SDK for Ruby API Referenz.

Für eine vollständige Liste von AWS SDK Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **GetBucketEncryption** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `GetBucketEncryption`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Get and print the encryption settings of a bucket.
/// </summary>
/// <param name="bucketName">Name of the bucket.</param>
/// <returns>Async task.</returns>
public static async Task GetEncryptionSettings(string bucketName)
{
    // Check and print the bucket encryption settings.
    Console.WriteLine($"Getting encryption settings for bucket
{bucketName}.");

    try
    {
        var settings =
            await _s3Client.GetBucketEncryptionAsync(
                new GetBucketEncryptionRequest() { BucketName =
bucketName });

        foreach (var encryptionSettings in
settings?.ServerSideEncryptionConfiguration?.ServerSideEncryptionRules!)
        {
            Console.WriteLine(
                $"{\tAlgorithm:
{encryptionSettings.ServerSideEncryptionByDefault.ServerSideEncryptionAlgorithm}");
            Console.WriteLine(
                $"{\tKey:
{encryptionSettings.ServerSideEncryptionByDefault.ServerSideEncryptionKeyManagementServi
                }
            }
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine(ex.ErrorCode == "InvalidBucketName"
```



```
        ? $"Bucket {bucketName} was not found."  
        : $"Unable to get bucket encryption for bucket {bucketName},  
        {ex.Message}");  
    }  
}
```

- API-Einheiten finden Sie [GetBucketEncryption](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um die serverseitige Verschlüsselungskonfiguration für einen Bucket abzurufen

Im folgenden `get-bucket-encryption` Beispiel wird die serverseitige Verschlüsselungskonfiguration für den Bucket `my-bucket`

```
aws s3api get-bucket-encryption \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "ServerSideEncryptionConfiguration": {  
    "Rules": [  
      {  
        "ApplyServerSideEncryptionByDefault": {  
          "SSEAlgorithm": "AES256"  
        }  
      }  
    ]  
  }  
}
```

- API-Einheiten finden Sie unter [GetBucketEncryption](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt alle serverseitigen Verschlüsselungsregeln zurück, die dem angegebenen Bucket zugeordnet sind.

```
Get-S3BucketEncryption -BucketName 's3casetestbucket'
```

- API-Einheiten finden Sie [GetBucketEncryption](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketInventoryConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketInventoryConfiguration`.

CLI

AWS CLI

Um die Inventarkonfiguration für einen Bucket abzurufen

Im folgenden `get-bucket-inventory-configuration` Beispiel wird die Inventarkonfiguration für den angegebenen Bucket mit der ID 1 abgerufen.

```
aws s3api get-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1
```

Ausgabe:

```
{  
  "InventoryConfiguration": {  
    "IsEnabled": true,  
  }  
}
```

```
    "Destination": {
      "S3BucketDestination": {
        "Format": "ORC",
        "Bucket": "arn:aws:s3:::my-bucket",
        "AccountId": "123456789012"
      }
    },
    "IncludedObjectVersions": "Current",
    "Id": "1",
    "Schedule": {
      "Frequency": "Weekly"
    }
  }
}
```

- API-Einheiten finden Sie unter [GetBucketInventoryConfiguration](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die Details des Inventars mit dem Namen 'testinventory' für den angegebenen S3-Bucket zurück.

```
Get-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
'testinventory'
```

- API-Einheiten finden Sie unter [GetBucketInventoryConfiguration](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketLifecycleConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketLifecycleConfiguration`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Returns a configuration object for the supplied bucket name.
/// </summary>
/// <param name="client">The S3 client object used to call
/// the GetLifecycleConfigurationAsync method.</param>
/// <param name="bucketName">The name of the S3 bucket for which a
/// configuration will be created.</param>
/// <returns>Returns a new LifecycleConfiguration object.</returns>
public static async Task<LifecycleConfiguration>
RetrieveLifecycleConfigAsync(IAmazonS3 client, string bucketName)
{
    var request = new GetLifecycleConfigurationRequest()
    {
        BucketName = bucketName,
    };
    var response = await client.GetLifecycleConfigurationAsync(request);
    var configuration = response.Configuration;
    return configuration;
}
```

- API-Einheiten finden Sie [GetBucketLifecycleConfiguration](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Der folgende Befehl ruft die Lebenszykluskonfiguration für einen Bucket mit dem Namen my-bucket ab:

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Ausgabe:

```
{
  "Rules": [
    {
      "ID": "Move rotated logs to Glacier",
      "Prefix": "rotated/",
      "Status": "Enabled",
      "Transitions": [
        {
          "Date": "2015-11-10T00:00:00.000Z",
          "StorageClass": "GLACIER"
        }
      ]
    },
    {
      "Status": "Enabled",
      "Prefix": "",
      "NoncurrentVersionTransitions": [
        {
          "NoncurrentDays": 0,
          "StorageClass": "GLACIER"
        }
      ],
      "ID": "Move old versions to Glacier"
    }
  ]
}
```

- API-Einheiten finden Sie unter [GetBucketLifecycleConfiguration](#) AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
            that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def get_lifecycle_configuration(self):
        """
        Get the lifecycle configuration of the bucket.

        :return: The lifecycle rules of the specified bucket.
        """
        try:
            config = self.bucket.LifecycleConfiguration()
            logger.info(
                "Got lifecycle rules %s for bucket '%s'.",
                config.rules,
                self.bucket.name,
            )
        except:
            logger.exception(
                "Couldn't get lifecycle rules for bucket '%s'.", self.bucket.name
            )
            raise
        else:
```

```
return config.rules
```

- API-Einheiten finden Sie [GetBucketLifecycleConfiguration](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketLocation** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketLocation`.

CLI

AWS CLI

Der folgende Befehl ruft die Speicherortbeschränkung für einen Bucket mit dem Namen `abmy-bucket`, falls eine Einschränkung existiert:

```
aws s3api get-bucket-location --bucket my-bucket
```

Ausgabe:

```
{
  "LocationConstraint": "us-west-2"
}
```

- API-Einheiten finden Sie unter [GetBucketLocation](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die Standortbeschränkung für den Bucket `'s3testbucket'` zurück, falls eine Einschränkung existiert.

```
Get-S3BucketLocation -BucketName 's3testbucket'
```

Ausgabe:

```
Value
-----
ap-south-1
```

- API-Einheiten finden Sie unter [GetBucketLocation](#) AWS Tools for PowerShell Cmdlet-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
async fn show_buckets(
    strict: bool,
    client: &Client,
    region: BucketLocationConstraint,
) -> Result<(), S3ExampleError> {
    let mut buckets = client.list_buckets().into_paginator().send();

    let mut num_buckets = 0;
    let mut in_region = 0;

    while let Some(Ok(output)) = buckets.next().await {
        for bucket in output.buckets() {
            num_buckets += 1;
            if strict {
                let r = client
                    .get_bucket_location()
                    .bucket(bucket.name().unwrap_or_default())
                    .send()
                    .await?;

                if r.location_constraint() == Some(&region) {
                    println!("{}", bucket.name().unwrap_or_default());
                }
            }
        }
    }
}
```



```
        in_region += 1;
    }
} else {
    println!("{}", bucket.name().unwrap_or_default());
}
}

println!();
if strict {
    println!(
        "Found {} buckets in the {} region out of a total of {} buckets.",
        in_region, region, num_buckets
    );
} else {
    println!("Found {} buckets in all regions.", num_buckets);
}

Ok(())
}
```

- API Einzelheiten finden Sie [GetBucketLocation](#) unter AWS SDK als API Referenz auf Rust.

Für eine vollständige Liste von AWS SDK Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **GetBucketLogging** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `GetBucketLogging`.

CLI

AWS CLI

Um den Logging-Status für einen Bucket abzurufen

Im folgenden `get-bucket-logging` Beispiel wird der Logging-Status für den angegebenen Bucket abgerufen.

```
aws s3api get-bucket-logging \
```

```
--bucket my-bucket
```

Ausgabe:

```
{
  "LoggingEnabled": {
    "TargetPrefix": "",
    "TargetBucket": "my-bucket-logs"
  }
}
```

- API-Einheiten finden Sie unter [GetBucketLogging](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt den Logging-Status für den angegebenen Bucket zurück.

```
Get-S3BucketLogging -BucketName 's3testbucket'
```

Ausgabe:

TargetBucketName	Grants	TargetPrefix
testbucket1	{}	testprefix

- API-Einheiten finden Sie [GetBucketLogging](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketMetricsConfiguration** mit einem AWS SDK oder CLI.

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketMetricsConfiguration`.

CLI

AWS CLI

Um die Metrikkonfiguration für einen Bucket mit einer bestimmten ID abzurufen

Im folgenden `get-bucket-metrics-configuration` Beispiel wird die Metrikkonfiguration für den angegebenen Bucket und die angegebene ID angezeigt.

```
aws s3api get-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Ausgabe:

```
{  
  "MetricsConfiguration": {  
    "Filter": {  
      "Prefix": "logs"  
    },  
    "Id": "123"  
  }  
}
```

- API-Einheiten finden Sie [GetBucketMetricsConfiguration](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die Details zum Metrikfilter mit dem Namen 'testfilter' für den angegebenen S3-Bucket zurück.

```
Get-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId  
'testfilter'
```

- API-Einheiten finden Sie unter [GetBucketMetricsConfiguration](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDKEntwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **GetBucketNotification** mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen die VerwendungGetBucketNotification.

CLI

AWS CLI

Der folgende Befehl ruft die Benachrichtigungskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-notification --bucket my-bucket
```

Ausgabe:

```
{
  "TopicConfiguration": {
    "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
    "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNL",
    "Event": "s3:ObjectCreated:*",
    "Events": [
      "s3:ObjectCreated:*"
    ]
  }
}
```

- APIEinzelheiten finden Sie unter [GetBucketNotification](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die Benachrichtigungskonfiguration des angegebenen Buckets abgerufen

```
Get-S3BucketNotification -BucketName kt-tools | select -ExpandProperty
TopicConfigurations
```

Ausgabe:

```
Id      Topic
--      -
mimo   arn:aws:sns:eu-west-1:123456789012:topic-1
```

- API-Einheiten finden Sie unter [GetBucketNotification](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketPolicy** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketPolicy`.

C++

SDK für C++

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::getBucketPolicy(const Aws::String &bucketName,
                                 const Aws::S3::S3ClientConfiguration
                                 &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::GetBucketPolicyRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketPolicyOutcome outcome =
        s3Client.GetBucketPolicy(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: getBucketPolicy: "
```

```

        << err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        Aws::StringStream policy_stream;
        Aws::String line;

        outcome.GetResult().GetPolicy() >> line;
        policy_stream << line;

        std::cout << "Retrieve the policy for bucket '" << bucketName << "':\n\n"
<<
        policy_stream.str() << std::endl;
    }

    return outcome.IsSuccess();
}

```

- API Einzelheiten finden Sie [GetBucketPolicy](#) unter AWS SDK for C++ API Referenz.

CLI

AWS CLI

Der folgende Befehl ruft die Bucket-Richtlinie für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-policy --bucket my-bucket
```

Ausgabe:

```

{
  "Policy": "{\"Version\":\"2008-10-17\",\"Statement\": [{\"Sid\":\"\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/*\"}, {\"Sid\":\"\", \"Effect\": \"Deny\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/secret/*\"} ]}"
}

```

Get and Put a Bucket Das policy The folgende Beispiel zeigt, wie Sie eine Amazon S3 S3-Bucket-Richtlinie herunterladen, Änderungen an der Datei vornehmen und dann die geänderte

Bucket-Richtlinie anwenden können. `put-bucket-policy` Um die Bucket-Richtlinie in eine Datei herunterzuladen, können Sie Folgendes ausführen:

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
policy.json
```

Anschließend können Sie die `policy.json` Datei nach Bedarf ändern. Schließlich können Sie diese geänderte Richtlinie wieder auf den S3-Bucket anwenden, indem Sie Folgendes ausführen:

`policy.json` Datei nach Bedarf. Schließlich können Sie diese geänderte Richtlinie wieder auf den S3-Bucket anwenden, indem Sie Folgendes ausführen:

Datei nach Bedarf. Schließlich können Sie diese geänderte Richtlinie wieder auf den S3-Bucket anwenden, indem Sie Folgendes ausführen:

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- API-Einheiten finden Sie [GetBucketPolicy](#) unter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetBucketPolicyRequest;
import software.amazon.awssdk.services.s3.model.GetBucketPolicyResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 */
```

```
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/

public class GetBucketPolicy {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName>

            Where:
                bucketName - The Amazon S3 bucket to get the policy from.
            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        System.out.format("Getting policy for bucket: \"%s\"\n\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        String polText = getPolicy(s3, bucketName);
        System.out.println("Policy Text: " + polText);
        s3.close();
    }

    public static String getPolicy(S3Client s3, String bucketName) {
        String policyText;
        System.out.format("Getting policy for bucket: \"%s\"\n\n", bucketName);
        GetBucketPolicyRequest policyReq = GetBucketPolicyRequest.builder()
            .bucket(bucketName)
            .build();

        try {
            GetBucketPolicyResponse policyRes = s3.getBucketPolicy(policyReq);
            policyText = policyRes.policy();
        }
    }
}
```



```
        return policyText;

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return "";
}
}
```

- API-Einheiten finden Sie [GetBucketPolicy](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Rufen Sie die Bucket-Richtlinie ab.

```
import { GetBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
    const command = new GetBucketPolicyCommand({
        Bucket: "test-bucket",
    });

    try {
        const { Policy } = await client.send(command);
        console.log(JSON.parse(Policy));
    } catch (err) {
        console.error(err);
    }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [GetBucketPolicy](#) in AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun getPolicy(bucketName: String): String? {
    println("Getting policy for bucket $bucketName")

    val request =
        GetBucketPolicyRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val policyRes = s3.getBucketPolicy(request)
        return policyRes.policy
    }
}
```

- API-Einheiten finden Sie [GetBucketPolicy](#) unter AWS SDK als API Kotlin-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die Bucket-Richtlinie aus, die dem angegebenen S3-Bucket zugeordnet ist.

```
Get-S3BucketPolicy -BucketName 's3testbucket'
```

- API-Einheiten finden Sie [GetBucketPolicy](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
            that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def get_policy(self):
        """
        Get the security policy of the bucket.

        :return: The security policy of the specified bucket, in JSON format.
        """
        try:
            policy = self.bucket.Policy()
            logger.info(
                "Got policy %s for bucket '%s'.", policy.policy, self.bucket.name
            )
        except ClientError:
```

```
        logger.exception("Couldn't get policy for bucket '%s'.",
self.bucket.name)
        raise
    else:
        return json.loads(policy.policy)
```

- API-Einheiten finden Sie [GetBucketPolicy](#) unter AWS SDK für Python (Boto3) API - Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object
  # configured with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  # Gets the policy of a bucket.
  #
  # @return [Aws::S3::GetBucketPolicyOutput, nil] The current bucket policy.
  def get_policy
    policy = @bucket_policy.data.policy
    policy.respond_to?(:read) ? policy.read : policy
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get the policy for #{@bucket_policy.bucket.name}. Here's why:
#{e.message}"
    nil
  end
end
```

```
end  
  
end
```

- API-Einheiten finden Sie [GetBucketPolicy](#) unter AWS SDK for Ruby API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketPolicyStatus** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketPolicyStatus`.

CLI

AWS CLI

Um den Richtlinienstatus für einen Bucket abzurufen, der angibt, ob der Bucket öffentlich ist

Im folgenden `get-bucket-policy-status` Beispiel wird der Richtlinienstatus für den Bucket `my-bucket` abgerufen.

```
aws s3api get-bucket-policy-status \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "PolicyStatus": {  
    "IsPublic": false  
  }  
}
```

- API-Einheiten finden Sie unter [GetBucketPolicyStatus](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt den Richtlinienstatus für den angegebenen S3-Bucket zurück und gibt an, ob der Bucket öffentlich ist.

```
Get-S3BucketPolicyStatus -BucketName 's3casetestbucket'
```

- API-Einheiten finden Sie [GetBucketPolicyStatus](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketReplication** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketReplication`.

CLI

AWS CLI

Der folgende Befehl ruft die Replikationskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-replication --bucket my-bucket
```

Ausgabe:

```
{
  "ReplicationConfiguration": {
    "Rules": [
      {
        "Status": "Enabled",
        "Prefix": "",
        "Destination": {
          "Bucket": "arn:aws:s3:::my-bucket-backup",
          "StorageClass": "STANDARD"
        }
      }
    ]
  }
}
```

```
        },
        "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
    }
],
"Role": "arn:aws:iam::123456789012:role/s3-replication-role"
}
}
```

- API-Einheiten finden Sie unter [GetBucketReplication](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Gibt die Informationen zur Replikationskonfiguration zurück, die für den Bucket mit dem Namen „mybucket“ festgelegt wurden.

```
Get-S3BucketReplication -BucketName mybucket
```

- Einzelheiten finden Sie unter API [GetBucketReplication](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketRequestPayment** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketRequestPayment`.

CLI

AWS CLI

Um die Konfiguration der Anforderungszahlung für einen Bucket abzurufen

Im folgenden `get-bucket-request-payment` Beispiel wird die Konfiguration für Zahlungen durch den Antragsteller für den angegebenen Bucket abgerufen.

```
aws s3api get-bucket-request-payment \
```

```
--bucket my-bucket
```

Ausgabe:

```
{  
  "Payer": "BucketOwner"  
}
```

- API-Einheiten finden Sie unter [GetBucketRequestPayment](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Gibt die Konfiguration der Anforderungszahlung für den Bucket mit dem Namen „mybucket“ zurück. Standardmäßig zahlt der Bucket-Besitzer für Downloads aus dem Bucket.

```
Get-S3BucketRequestPayment -BucketName mybucket
```

- API-Einheiten finden Sie [GetBucketRequestPayment](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketTagging** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketTagging`.

CLI

AWS CLI

Der folgende Befehl ruft die Tagging-Konfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-tagging --bucket my-bucket
```


Ausgabe:

```
{
  "TagSet": [
    {
      "Value": "marketing",
      "Key": "organization"
    }
  ]
}
```

- API-Einheiten finden Sie unter [GetBucketTagging](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt alle Tags zurück, die dem angegebenen Bucket zugeordnet sind.

```
Get-S3BucketTagging -BucketName 's3casetestbucket'
```

- API-Einheiten finden Sie [GetBucketTagging](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketVersioning** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketVersioning`.

CLI

AWS CLI

Der folgende Befehl ruft die Versionierungskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Ausgabe:

```
{
  "Status": "Enabled"
}
```

- API-Einheiten finden Sie unter [GetBucketVersioning](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt den Status der Versionierung in Bezug auf den angegebenen Bucket zurück.

```
Get-S3BucketVersioning -BucketName 's3testbucket'
```

- API-Einheiten finden Sie unter [GetBucketVersioning](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetBucketWebsite** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetBucketWebsite`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// Get the website configuration.
```

```

        GetBucketWebsiteRequest getRequest = new
GetBucketWebsiteRequest()
        {
            BucketName = bucketName,
        };
        GetBucketWebsiteResponse getResponse = await
client.GetBucketWebsiteAsync(getRequest);
        Console.WriteLine($"Index document:
{getResponse.WebsiteConfiguration.IndexDocumentSuffix}");
        Console.WriteLine($"Error document:
{getResponse.WebsiteConfiguration.ErrorDocument}");

```

- API-Einheiten finden Sie [GetBucketWebsite](#) unter AWS SDK for .NET API-Referenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

bool AwsDoc::S3::getWebsiteConfig(const Aws::String &bucketName,
                                   const Aws::S3::S3ClientConfiguration
&clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::GetBucketWebsiteRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketWebsiteOutcome outcome =
        s3Client.GetBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();

        std::cerr << "Error: GetBucketWebsite: "
            << err.GetMessage() << std::endl;
    }
}

```

```
    } else {
        Aws::S3::Model::GetBucketWebsiteResult websiteResult =
outcome.GetResult();

        std::cout << "Success: GetBucketWebsite: "
            << std::endl << std::endl
            << "For bucket '" << bucketName << "':"
            << std::endl
            << "Index page : "
            << websiteResult.GetIndexDocument().GetSuffix()
            << std::endl
            << "Error page: "
            << websiteResult.GetErrorDocument().GetKey()
            << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API-Einheiten finden Sie [GetBucketWebsite](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl ruft die statische Website-Konfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-website --bucket my-bucket
```

Ausgabe:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

- API-Einheiten finden Sie unter [GetBucketWebsite](#) AWS CLI Befehlsreferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Rufen Sie die Website-Konfiguration ab.

```
import { GetBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketWebsiteCommand({
    Bucket: "test-bucket",
  });

  try {
    const { ErrorDocument, IndexDocument } = await client.send(command);
    console.log(
      `Your bucket is set up to host a website. It has an error document:`,
      `${ErrorDocument.Key}, and an index document: ${IndexDocument.Suffix}.`,
    );
  } catch (err) {
    console.error(err);
  }
};
```

- API-Einheiten finden Sie [GetBucketWebsite](#) unter AWS SDK for JavaScript API Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die Details der statischen Website-Konfigurationen des angegebenen S3-Buckets zurück.

```
Get-S3BucketWebsite -BucketName 's3testbucket'
```

- API-Einheiten finden Sie [GetBucketWebsite](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetObject** mit einem AWS SDK-Koder CLI

Die folgenden Codebeispiele zeigen die Verwendung `GetObject`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erlernen Sie die Grundlagen](#)
- [Abrufen eines Objekts aus einem Bucket abrufen, wenn es geändert wurde](#)
- [Rufen Sie ein Objekt von einem Access Point mit mehreren Regionen ab](#)
- [Erste Schritte mit der Verschlüsselung](#)
- [Verfolgen Sie Uploads und Downloads](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Shows how to download an object from an Amazon S3 bucket to the
/// local computer.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket where the object is
/// currently stored.</param>
/// <param name="objectName">The name of the object to download.</param>
/// <param name="filePath">The path, including filename, where the
/// downloaded object will be stored.</param>
/// <returns>A boolean value indicating the success or failure of the
/// download process.</returns>
public static async Task<bool> DownloadObjectFromBucketAsync(
    IAmazonS3 client,
    string bucketName,
    string objectName,
    string filePath)
{
    // Create a GetObject request
    var request = new GetObjectRequest
    {
        BucketName = bucketName,
        Key = objectName,
    };

    // Issue request and remember to dispose of the response
    using GetObjectResponse response = await
client.GetObjectAsync(request);

    try
    {
        // Save object to local file
        await response.WriteResponseStreamToFileAsync($"{filePath}\
\{objectName}", true, Cancellation.Token.None);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error saving {objectName}: {ex.Message}");
        return false;
    }
}
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#     $1 - The name of the bucket to download the object from.
#     $2 - The path and file name to store the downloaded bucket.
#     $3 - The key (name) of the object in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function download_object_from_bucket() {
    local bucket_name=$1
    local destination_file_name=$2
    local object_name=$3
```



```
local response

response=$(aws s3api get-object \
  --bucket "$bucket_name" \
  --key "$object_name" \
  "$destination_file_name")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
  errecho "ERROR: AWS reports put-object operation failed.\n$response"
  return 1
fi
}
```

- API-Einheiten finden Sie [GetObject](#) unter AWS CLI Befehlsreferenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::getObject(const Aws::String &objectKey,
                           const Aws::String &fromBucket,
                           const Aws::S3::S3ClientConfiguration &clientConfig) {
  Aws::S3::S3Client client(clientConfig);

  Aws::S3::Model::GetObjectRequest request;
  request.SetBucket(fromBucket);
  request.SetKey(objectKey);

  Aws::S3::Model::GetObjectOutcome outcome =
    client.GetObject(request);

  if (!outcome.IsSuccess()) {
    const Aws::S3::S3Error &err = outcome.GetError();
    std::cerr << "Error: getObject: " <<
```

```
        err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        std::cout << "Successfully retrieved '" << objectKey << "' from '"
        << fromBucket << "'." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Im folgenden Beispiel wird der `get-object` Befehl verwendet, um ein Objekt von Amazon S3 herunterzuladen:

```
aws s3api get-object --bucket text-content --key dir/
my_images.tar.bz2 my_images.tar.bz2
```

Beachten Sie, dass der Outfile-Parameter ohne einen Optionsnamen wie „--outfile“ angegeben wird. Der Name der Ausgabedatei muss der letzte Parameter im Befehl sein.

Das folgende Beispiel zeigt die Verwendung von `--range`, um einen bestimmten Bytebereich von einem Objekt herunterzuladen. Beachten Sie, dass den Bytebereichen das Präfix „bytes=“ vorangestellt werden muss:

```
aws s3api get-object --bucket text-content --key dir/my_data --
range bytes=8888-9999 my_data_range
```

Weitere Informationen zum Abrufen von Objekten finden Sie unter Getting Objects im Amazon S3 Developer Guide.

- API-Einheiten finden Sie unter [GetObject](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// DownloadFile gets an object from a bucket and stores it in a local file.
func (basics BucketBasics) DownloadFile(bucketName string, objectKey string,
    fileName string) error {
    result, err := basics.S3Client.GetObject(context.TODO(), &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't get object %v:%v. Here's why: %v\n", bucketName,
            objectKey, err)
        return err
    }
    defer result.Body.Close()
    file, err := os.Create(fileName)
    if err != nil {
        log.Printf("Couldn't create file %v. Here's why: %v\n", fileName, err)
        return err
    }
    defer file.Close()
```

```
body, err := io.ReadAll(result.Body)
if err != nil {
    log.Printf("Couldn't read object body from %v. Here's why: %v\n", objectKey,
err)
}
_, err = file.Write(body)
return err
}
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Lesen Sie Daten als Byte-Array mit einem [S3Client](#).

```
/**
 * Asynchronously retrieves the bytes of an object from an Amazon S3 bucket
and writes them to a local file.
 *
 * @param bucketName the name of the S3 bucket containing the object
 * @param keyName the key (or name) of the S3 object to retrieve
 * @param path the local file path where the object's bytes will be written
 * @return a {@link CompletableFuture} that completes when the object bytes
have been written to the local file
 */
public CompletableFuture<Void> getObjectBytesAsync(String bucketName, String
keyName, String path) {
    GetObjectRequest objectRequest = GetObjectRequest.builder()
        .key(keyName)
        .bucket(bucketName)
        .build();
```

```
CompletableFuture<ResponseBytes<GetObjectResponse>> response =
getAsyncClient().getObject(objectRequest, AsyncResponseTransformer.toBytes());
return response.thenAccept(objectBytes -> {
    try {
        byte[] data = objectBytes.asByteArray();
        Path filePath = Paths.get(path);
        Files.write(filePath, data);
        logger.info("Successfully obtained bytes from an S3 object");
    } catch (IOException ex) {
        throw new RuntimeException("Failed to write data to file", ex);
    }
}).whenComplete((resp, ex) -> {
    if (ex != null) {
        throw new RuntimeException("Failed to get object bytes from S3",
ex);
    }
});
}
```

Verwenden Sie ein [S3 TransferManager](#), um [ein Objekt in einem S3-Bucket in eine lokale Datei herunterzuladen](#). Sehen Sie sich die [vollständige Datei](#) an und [testen](#) Sie sie.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadFileRequest;
import software.amazon.awssdk.transfer.s3.model.FileDownload;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;

import java.io.IOException;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.UUID;

    public Long downloadFile(S3TransferManager transferManager, String
bucketName,
```

```

        String key, String downloadedFilePath) {
    DownloadFileRequest downloadFileRequest = DownloadFileRequest.builder()
        .getObjectRequest(b -> b.bucket(bucketName).key(key))
        .destination(Paths.get(downloadedFilePath))
        .build();

    FileDownload downloadFile =
transferManager.downloadFile(downloadFileRequest);

    CompletedFileDownload downloadResult =
downloadFile.completionFuture().join();
    logger.info("Content length [{}]",
downloadResult.response().contentType());
    return downloadResult.response().contentType();
}

```

Lesen Sie Tags, die zu einem Objekt gehören, mit einem [S3Client](#).

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.Tag;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetObjectTags {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName> <keyName>\s

```

```
        Where:
            bucketName - The Amazon S3 bucket name.\s
            keyName - A key name that represents the object.\s
            """";

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String bucketName = args[0];
    String keyName = args[1];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    listTags(s3, bucketName, keyName);
    s3.close();
}

public static void listTags(S3Client s3, String bucketName, String keyName) {
    try {
        GetObjectTaggingRequest getTaggingRequest = GetObjectTaggingRequest
            .builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        GetObjectTaggingResponse tags =
s3.getObjectTagging(getTaggingRequest);
        List<Tag> tagSet = tags.tagSet();
        for (Tag tag : tagSet) {
            System.out.println(tag.key());
            System.out.println(tag.value());
        }

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

Rufen Sie mit einem [S3Client](#) a URL für ein Objekt ab.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetUrlRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.net.URL;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */

public class GetObjectUrl {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <keyName>\s

            Where:
                bucketName - The Amazon S3 bucket name.
                keyName - A key name that represents the object.\s
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();
```



```
        getURL(s3, bucketName, keyName);
        s3.close();
    }

    public static void getURL(S3Client s3, String bucketName, String keyName) {
        try {
            GetUrlRequest request = GetUrlRequest.builder()
                .bucket(bucketName)
                .key(keyName)
                .build();

            URL url = s3.utilities().getUrl(request);
            System.out.println("The URL for " + keyName + " is " + url);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Rufen Sie ein Objekt mithilfe des S3Presigner-Client-Objekts mit einem [S3Client](#) ab.

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.time.Duration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import
    software.amazon.awssdk.services.s3.presigner.model.GetObjectPresignRequest;
import
    software.amazon.awssdk.services.s3.presigner.model.PresignedGetObjectRequest;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import software.amazon.awssdk.utils.IoUtils;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class GetObjectPresignedUrl {
    public static void main(String[] args) {
        final String USAGE = ""

            Usage:
                <bucketName> <keyName>\s

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - A key name that represents a text file.\s
            """;

        if (args.length != 2) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Presigner presigner = S3Presigner.builder()
            .region(region)
            .build();

        getPresignedUrl(presigner, bucketName, keyName);
        presigner.close();
    }

    public static void getPresignedUrl(S3Presigner presigner, String bucketName,
        String keyName) {
        try {
            GetObjectRequest getObjectRequest = GetObjectRequest.builder()
                .bucket(bucketName)
                .key(keyName)
                .build();

            GetObjectPresignRequest getObjectPresignRequest =
                GetObjectPresignRequest.builder()
```

```
        .signatureDuration(Duration.ofMinutes(60))
        .getObjectRequest(getObjectRequest)
        .build();

    PresignedGetObjectRequest presignedGetObjectRequest =
presigner.presignGetObject(getObjectPresignRequest);
    String theUrl = presignedGetObjectRequest.url().toString();
    System.out.println("Presigned URL: " + theUrl);
    HttpURLConnection connection = (HttpURLConnection)
presignedGetObjectRequest.url().openConnection();
    presignedGetObjectRequest.httpRequest().headers().forEach((header,
values) -> {
        values.forEach(value -> {
            connection.addRequestProperty(header, value);
        });
    });

    // Send any request payload that the service needs (not needed when
// isBrowserExecutable is true).
    if (presignedGetObjectRequest.signedPayload().isPresent()) {
        connection.setDoOutput(true);

        try (InputStream signedPayload =
presignedGetObjectRequest.signedPayload().get().asInputStream();
            OutputStream httpOutputStream =
connection.getOutputStream()) {
            IoUtils.copy(signedPayload, httpOutputStream);
        }
    }

    // Download the result of executing the request.
    try (InputStream content = connection.getInputStream()) {
        System.out.println("Service returned response: ");
        IoUtils.copy(content, System.out);
    }

} catch (S3Exception | IOException e) {
    e.printStackTrace();
}
}
```

Ruft ein Objekt ab, indem Sie ein ResponseTransformer Objekt und S3Client verwenden.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.core.sync.ResponseTransformer;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class GetDataResponseTransformer {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <keyName> <path>

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - The key name.\s
                path - The path where the file is written to.\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
```

```
String path = args[2];
Region region = Region.US_EAST_1;
S3Client s3 = S3Client.builder()
    .region(region)
    .build();

getObjectBytes(s3, bucketName, keyName, path);
s3.close();
}

public static void getObjectBytes(S3Client s3, String bucketName, String
keyName, String path) {
    try {
        GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        ResponseBytes<GetObjectResponse> objectBytes =
s3.getObject(objectRequest, ResponseTransformer.toBytes());
        byte[] data = objectBytes.asByteArray();

        // Write the data to a local file.
        File myFile = new File(path);
        OutputStream os = new FileOutputStream(myFile);
        os.write(data);
        System.out.println("Successfully obtained bytes from an S3 object");
        os.close();

    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- APIEinzelheiten finden Sie unter [GetObject](#) AWS SDK for Java 2.x APIReferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Laden Sie das Objekt herunter.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
  });

  try {
    const response = await client.send(command);
    // The Body object also has 'transformToByteArray' and 'transformToWebStream'
    methods.
    const str = await response.Body.transformToString();
    console.log(str);
  } catch (err) {
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [GetObject](#) in AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun getObjectBytes(
    bucketName: String,
    keyName: String,
    path: String,
) {
    val request =
        GetObjectRequest {
            key = keyName
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.getObject(request) { resp ->
            val myFile = File(path)
            resp.body?.writeToFile(myFile)
            println("Successfully read $keyName from $bucketName")
        }
    }
}
```

- API Einzelheiten finden Sie [GetObject](#) unter AWS SDK als API Kotlin-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Rufen Sie ein Objekt ab.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $file = $this->s3client->getObject([
        'Bucket' => $this->bucketName,
        'Key' => $fileName,
    ]);
    $body = $file->get('Body');
    $body->rewind();
    echo "Downloaded the file and it begins with: {$body->read(26)}.\n";
} catch (Exception $exception) {
    echo "Failed to download $fileName from $this->bucketName with error:
" . $exception->getMessage();
    exit("Please fix error with file downloading before continuing.");
}
```

- APIEinzelheiten finden Sie [GetObject](#)unter AWS SDK for PHP APIReferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl ruft das Element "sample.txt" aus dem Bucket „test-files“ ab und speichert es in einer Datei mit dem Namen "local-sample.txt" am aktuellen Speicherort. Die Datei "local-sample.txt" muss nicht existieren, bevor dieser Befehl aufgerufen wird.

```
Read-S3Object -BucketName test-files -Key sample.txt -File local-sample.txt
```


Beispiel 2: Dieser Befehl ruft das virtuelle Verzeichnis "DIR" aus dem Bucket „test-files“ ab und speichert es in einem Ordner mit dem Namen „Local-DIR“ am aktuellen Speicherort. Der Ordner „Local-DIR“ muss nicht existieren, bevor dieser Befehl aufgerufen wird.

```
Read-S3Object -BucketName test-files -KeyPrefix DIR -Folder Local-DIR
```

Beispiel 3: Lädt alle Objekte mit Schlüsseln, die auf '.json' enden, aus Buckets mit 'config' im Bucket-Namen in Dateien im angegebenen Ordner herunter. Die Objektschlüssel werden verwendet, um die Dateinamen festzulegen.

```
Get-S3Bucket | ? { $_.BucketName -like '*config*' } | Get-S3Object | ? { $_.Key -like '*.json' } | Read-S3Object -Folder C:\ConfigObjects
```

- API-Einheiten finden Sie unter [GetObject](#) AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
                               that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def get(self):
        """
```

```
Gets the object.

:return: The object data in bytes.
"""
try:
    body = self.object.get()["Body"].read()
    logger.info(
        "Got object '%s' from bucket '%s'.",
        self.object.key,
        self.object.bucket_name,
    )
except ClientError:
    logger.exception(
        "Couldn't get object '%s' from bucket '%s'.",
        self.object.key,
        self.object.bucket_name,
    )
    raise
else:
    return body
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Rufen Sie ein Objekt ab.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetWrapper
  attr_reader :object
```

```
# @param object [Aws::S3::Object] An existing Amazon S3 object.
def initialize(object)
  @object = object
end

# Gets the object directly to a file.
#
# @param target_path [String] The path to the file where the object is
downloaded.
# @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
successful; otherwise nil.
def get_object(target_path)
  @object.get(response_target: target_path)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object.txt"
  target_path = "my-object-as-file.txt"

  wrapper = ObjectGetWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  obj_data = wrapper.get_object(target_path)
  return unless obj_data

  puts "Object #{object_key} (#{obj_data.content_length} bytes) downloaded to
#{target_path}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Rufen Sie ein Objekt ab und melden Sie seinen serverseitigen Verschlüsselungsstatus.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object
```

```
# @param object [Aws::S3::Object] An existing Amazon S3 object.
def initialize(object)
  @object = object
end

# Gets the object into memory.
#
# @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
successful; otherwise nil.
def get_object
  @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? "no" :
obj_data.server_side_encryption
  puts "Object #{object_key} uses #{encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- API Einzelheiten finden Sie [GetObject](#) unter AWS SDK for Ruby API Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
async fn get_object(client: Client, opt: Opt) -> Result<usize, S3ExampleError> {
    trace!("bucket:      {}", opt.bucket);
    trace!("object:      {}", opt.object);
    trace!("destination: {}", opt.destination.display());

    let mut file = File::create(opt.destination.clone()).map_err(|err| {
        S3ExampleError::new(format!(
            "Failed to initialize file for saving S3 download: {err:?}"
        ))
    })?;

    let mut object = client
        .get_object()
        .bucket(opt.bucket)
        .key(opt.object)
        .send()
        .await?;

    let mut byte_count = 0_usize;
    while let Some(bytes) = object.body.try_next().await.map_err(|err| {
        S3ExampleError::new(format!("Failed to read from S3 download stream:
{err:?}"))
    })? {
        let bytes_len = bytes.len();
        file.write_all(&bytes).map_err(|err| {
            S3ExampleError::new(format!(
                "Failed to write from S3 download stream to local file: {err:?}"
            ))
        })?;
        trace!("Intermediate write of {bytes_len}");
        byte_count += bytes_len;
    }
}
```

```
    Ok(byte_count)
}
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK als API-Referenz auf Rust.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
TRY.
    oo_result = lo_s3->getobject(           " oo_result is returned for
testing purposes. "
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key
    ).
    DATA(lv_object_data) = oo_result->get_body( ).
    MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK als SAP ABAP API-Referenz.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3

public func downloadFile(bucket: String, key: String, to: String) async
throws {
    let fileUrl = URL(fileURLWithPath: to).appendingPathComponent(key)

    let input = GetObjectInput(
        bucket: bucket,
        key: key
    )
    do {
        let output = try await client.getObject(input: input)

        guard let body = output.body else {
            throw HandlerError.getObjectBody("GetObjectInput missing body.")
        }

        guard let data = try await body.readData() else {
            read data.")
        }

        try data.write(to: fileUrl)
    }
    catch {
        print("ERROR: ", dump(error, name: "Downloading a file. "))
        throw error
    }
}
```

```
import AWSS3
```

```
public func readFile(bucket: String, key: String) async throws -> Data {
    let input = GetObjectInput(
        bucket: bucket,
        key: key
    )
    do {
        let output = try await client.getObject(input: input)

        guard let body = output.body else {
            throw HandlerError.getObjectBody("GetObjectInput missing body.")
        }

        guard let data = try await body.readData() else {
            throw HandlerError.readGetObjectBody("GetObjectInput unable to
read data.")
        }

        return data
    }
    catch {
        print("ERROR: ", dump(error, name: "Reading a file. "))
        throw error
    }
}
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK als API-Swift-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetObjectACL** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetObjectACL`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Zugriffskontrolllisten verwalten \(\) ACLs](#)

C++

SDK für C++

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::getObjectAcl(const Aws::String &bucketName,
                              const Aws::String &objectKey,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::GetObjectAclRequest request;
    request.SetBucket(bucketName);
    request.SetKey(objectKey);

    Aws::S3::Model::GetObjectAclOutcome outcome =
        s3Client.GetObjectAcl(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: getObjectAcl: "
                  << err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        Aws::Vector<Aws::S3::Model::Grant> grants =
            outcome.GetResult().GetGrants();

        for (auto it = grants.begin(); it != grants.end(); it++) {
            std::cout << "For object " << objectKey << ": "
                      << std::endl << std::endl;

            Aws::S3::Model::Grant grant = *it;
            Aws::S3::Model::Grantee grantee = grant.GetGrantee();

            if (grantee.TypeHasBeenSet()) {
                std::cout << "Type:          "
```

```

        << getGranteeTypeString(grantee.GetType()) <<
std::endl;
    }

    if (grantee.DisplayNameHasBeenSet()) {
        std::cout << "Display name: "
            << grantee.GetDisplayName() << std::endl;
    }

    if (grantee.EmailAddressHasBeenSet()) {
        std::cout << "Email address: "
            << grantee.GetEmailAddress() << std::endl;
    }

    if (grantee.IDHasBeenSet()) {
        std::cout << "ID: "
            << grantee.GetID() << std::endl;
    }

    if (grantee.URIHasBeenSet()) {
        std::cout << "URI: "
            << grantee.GetURI() << std::endl;
    }

    std::cout << "Permission: " <<
        getPermissionString(grant.GetPermission()) <<
        std::endl << std::endl;
    }
}

return outcome.IsSuccess();
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \param type: Type enumeration.
 \return String: Human-readable string
 */
Aws::String getGranteeTypeString(const Aws::S3::Model::Type &type) {
    switch (type) {
        case Aws::S3::Model::Type::AmazonCustomerByEmail:
            return "Email address of an AWS account";
        case Aws::S3::Model::Type::CanonicalUser:

```

```
        return "Canonical user ID of an AWS account";
    case Aws::S3::Model::Type::Group:
        return "Predefined Amazon S3 group";
    case Aws::S3::Model::Type::NOT_SET:
        return "Not set";
    default:
        return "Type unknown";
    }
}

//! Routine which converts a built-in type enumeration to a human-readable
string.
/*!
 \param permission: Permission enumeration.
 \return String: Human-readable string
 */
Aws::String getPermissionString(const Aws::S3::Model::Permission &permission) {
    switch (permission) {
        case Aws::S3::Model::Permission::FULL_CONTROL:
            return "Can read this object's data and its metadata, "
                "and read/write this object's permissions";
        case Aws::S3::Model::Permission::NOT_SET:
            return "Permission not set";
        case Aws::S3::Model::Permission::READ:
            return "Can read this object's data and its metadata";
        case Aws::S3::Model::Permission::READ_ACP:
            return "Can read this object's permissions";
        // case Aws::S3::Model::Permission::WRITE // Not applicable.
        case Aws::S3::Model::Permission::WRITE_ACP:
            return "Can write this object's permissions";
        default:
            return "Permission unknown";
    }
}
```

- APIEinzelheiten finden Sie [GetObjectAcl](#) unter AWS SDK for C++ APIReferenz.

CLI

AWS CLI

Der folgende Befehl ruft die Zugriffskontrollliste für ein Objekt in einem Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Ausgabe:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    },
    {
      "Grantee": {
        "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
      },
      "Permission": "READ"
    }
  ]
}
```

- API-Einheiten finden Sie unter [GetObjectAcl](#) AWS CLI Befehlsreferenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun getBucketACL(
    objectKey: String,
    bucketName: String,
) {
    val request =
        GetObjectAclRequest {
            bucket = bucketName
            key = objectKey
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.getObjectAcl(request)
        response.grants?.forEach { grant ->
            println("Grant permission is ${grant.permission}")
        }
    }
}
```

- API Einheiten finden Sie [GetObjectAcl](#) unter AWS SDK als API Kotlin-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
                               that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def get_acl(self):
        """
        Gets the ACL of the object.

        :return: The ACL of the object.
        """
        try:
            acl = self.object.Acl()
            logger.info(
                "Got ACL for object %s owned by %s.",
                self.object.key,
                acl.owner["DisplayName"],
            )
        except ClientError:
            logger.exception("Couldn't get ACL for object %s.", self.object.key)
            raise
        else:
            return acl
```

- API-Einheiten finden Sie [GetObjectAcl](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie `GetObjectAttributes` mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen die Verwendung `GetObjectAttributes`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Arbeiten Sie mit Amazon S3 S3-Objektintegrität](#)

C++

SDKfür C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// ! Routine which retrieves the hash value of an object stored in an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object is stored.
    \param key: The unique identifier (key) of the object within the S3 bucket.
    \param hashMethod: The hashing algorithm used to calculate the hash value of
the object.
    \param[out] hashData: The retrieved hash.
    \param[out] partHashes: The part hashes if available.
    \param client: The S3 client instance used to retrieve the object.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::retrieveObjectHash(const Aws::String &bucket, const Aws::String
&key,
                                   AwsDoc::S3::HASH_METHOD hashMethod,
                                   Aws::String &hashData,
                                   std::vector<Aws::String> *partHashes,
                                   const Aws::S3::S3Client &client) {
    Aws::S3::Model::GetObjectAttributesRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);

    if (hashMethod == MD5) {
        Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
```

```
attributes.push_back(Aws::S3::Model::ObjectAttributes::ETag);
request.SetObjectAttributes(attributes);

Aws::S3::Model::GetObjectAttributesOutcome outcome =
client.GetObjectAttributes(
    request);
if (outcome.IsSuccess()) {
    const Aws::S3::Model::GetObjectAttributesResult &result =
outcome.GetResult();
    hashData = result.GetETag();
} else {
    std::cerr << "Error retrieving object etag attributes." <<
        outcome.GetError().GetMessage() << std::endl;
    return false;
}
} else { // hashMethod != MD5
    Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
    attributes.push_back(Aws::S3::Model::ObjectAttributes::Checksum);
    request.SetObjectAttributes(attributes);

    Aws::S3::Model::GetObjectAttributesOutcome outcome =
client.GetObjectAttributes(
    request);
    if (outcome.IsSuccess()) {
        const Aws::S3::Model::GetObjectAttributesResult &result =
outcome.GetResult();
        switch (hashMethod) {
            case AwsDoc::S3::DEFAULT: // NOLINT(*-branch-clone)
                break; // Default is not supported.
#pragma clang diagnostic push
#pragma ide diagnostic ignored "UnreachableCode"
            case AwsDoc::S3::MD5:
                break; // MD5 is not supported.
#pragma clang diagnostic pop
            case AwsDoc::S3::SHA1:
                hashData = result.GetChecksum().GetChecksumSHA1();
                break;
            case AwsDoc::S3::SHA256:
                hashData = result.GetChecksum().GetChecksumSHA256();
                break;
            case AwsDoc::S3::CRC32:
                hashData = result.GetChecksum().GetChecksumCRC32();
                break;
            case AwsDoc::S3::CRC32C:
```



```

        hashData = result.GetChecksum().GetChecksumCRC32C();
        break;
    default:
        std::cerr << "Unknown hash method." << std::endl;
        return false;
    }
} else {
    std::cerr << "Error retrieving object checksum attributes." <<
        outcome.GetError().GetMessage() << std::endl;
    return false;
}

if (nullptr != partHashes) {
    attributes.clear();
    attributes.push_back(Aws::S3::Model::ObjectAttributes::ObjectParts);
    request.SetObjectAttributes(attributes);
    outcome = client.GetObjectAttributes(request);
    if (outcome.IsSuccess()) {
        const Aws::S3::Model::GetObjectAttributesResult &result =
outcome.GetResult();
        const Aws::Vector<Aws::S3::Model::ObjectPart> parts =
result.GetObjectParts().GetParts();
        for (const Aws::S3::Model::ObjectPart &part: parts) {
            switch (hashMethod) {
                case AwsDoc::S3::DEFAULT: // Default is not supported.
NOLINT(*-branch-clone)
                    break;
                case AwsDoc::S3::MD5: // MD5 is not supported.
                    break;
                case AwsDoc::S3::SHA1:
                    partHashes->push_back(part.GetChecksumSHA1());
                    break;
                case AwsDoc::S3::SHA256:
                    partHashes->push_back(part.GetChecksumSHA256());
                    break;
                case AwsDoc::S3::CRC32:
                    partHashes->push_back(part.GetChecksumCRC32());
                    break;
                case AwsDoc::S3::CRC32C:
                    partHashes->push_back(part.GetChecksumCRC32C());
                    break;
                default:
                    std::cerr << "Unknown hash method." << std::endl;
                    return false;
            }
        }
    }
}

```

```
        }
    }
} else {
    std::cerr << "Error retrieving object attributes for object
parts." <<
        outcome.GetError().GetMessage() << std::endl;
    return false;
}
}
}
return true;
}
```

- API-Einheiten finden Sie [GetObjectAttributes](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Ruft Metadaten von einem Objekt ab, ohne das Objekt selbst zurückzugeben

Im folgenden `get-object-attributes` Beispiel werden Metadaten aus dem Objekt abgerufen. `doc1.rtf`

```
aws s3api get-object-attributes \
  --bucket my-bucket \
  --key doc1.rtf \
  --object-attributes "StorageClass" "ETag" "ObjectSize"
```

Ausgabe:

```
{
  "LastModified": "2022-03-15T19:37:31+00:00",
  "VersionId": "IuCPjXTDzHNf1dAuitVBIKJpF2p1fg4P",
  "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",
  "StorageClass": "STANDARD",
  "ObjectSize": 405
}
```

Weitere Informationen finden Sie [GetObjectAttributes](#) in der Amazon S3 API S3-Referenz.

- API-Einheiten finden Sie [GetObjectAttributes](#) unter AWS CLI Befehlsreferenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetObjectLegalHold** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetObjectLegalHold`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Amazon S3 S3-Objekte sperren](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Get the legal hold details for an S3 object.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object legal hold details.</returns>
public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        }
    }
}
```

```
};

var response = await _amazonS3.GetObjectLegalHoldAsync(request);
Console.WriteLine($"Object legal hold for {objectKey} in
{bucketName}: " +
    $"Status: {response.LegalHold.Status}");
return response.LegalHold;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Unable to fetch legal hold: '{ex.Message}'");
    return new ObjectLockLegalHold();
}
}
```

- API-Einheiten finden Sie [GetObjectLegalHold](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Ruft den Status „Legal Hold“ eines Objekts ab

Im folgenden `get-object-legal-hold` Beispiel wird der Status Legal Hold für das angegebene Objekt abgerufen.

```
aws s3api get-object-legal-hold \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf
```

Ausgabe:

```
{
  "LegalHold": {
    "Status": "ON"
  }
}
```

- API-Einheiten finden Sie unter [GetObjectLegalHold](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
    S3Client    *s3.Client
    S3Manager  *manager.Uploader
}

// GetObjectLegalHold retrieves the legal hold status for an S3 object.
func (actor S3Actions) GetObjectLegalHold(ctx context.Context, bucket string, key
string, versionId string) (*types.ObjectLockLegalHoldStatus, error) {
    var status *types.ObjectLockLegalHoldStatus
    input := &s3.GetObjectLegalHoldInput{
        Bucket:    aws.String(bucket),
        Key:       aws.String(key),
        VersionId: aws.String(versionId),
    }

    output, err := actor.S3Client.GetObjectLegalHold(ctx, input)
    if err != nil {
        var noSuchKeyErr *types.NoSuchKey
        var apiErr *smithy.GenericAPIError
        if errors.As(err, &noSuchKeyErr) {
            log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
            err = noSuchKeyErr
        } else if errors.As(err, &apiErr) {
            switch apiErr.ErrorCode() {
            case "NoSuchObjectLockConfiguration":
                log.Printf("Object %s does not have an object lock configuration.\n", key)
                err = nil
            case "InvalidRequest":
```

```
    log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
    err = nil
}
}
} else {
    status = &output.LegalHold.Status
}

return status, err
}
```

- API-Einheiten finden Sie [GetObjectLegalHold](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// Get the legal hold details for an S3 object.
public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
    try {
        GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .build();

        GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
        System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
            ":\n\tStatus: " + response.legalHold().status());
        return response.legalHold();
    }
}
```

```
    } catch (S3Exception ex) {
        System.out.println("\tUnable to fetch legal hold: '" +
ex.getMessage() + "'");
    }

    return null;
}
```

- API-Einheiten finden Sie [GetObjectLegalHold](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import { fileURLToPath } from "url";
import { GetObjectLegalHoldCommand, S3Client } from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
    const command = new GetObjectLegalHoldCommand({
        Bucket: bucketName,
        Key: objectKey,
        // Optionally, you can provide additional parameters
        // ExpectedBucketOwner: "ACCOUNT_ID",
        // RequestPayer: "requester",
        // VersionId: "OBJECT_VERSION_ID",
    });

    try {
        const response = await client.send(command);
```

```
    console.log(`Legal Hold Status: ${response.LegalHold.Status}`);
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "amzn-s3-demo-bucket", "OBJECT_KEY");
}
```

- API Einzelheiten finden Sie [GetObjectLegalHold](#) unter AWS SDK for JavaScript API Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Verschaffen Sie einem Objekt eine gesetzliche Sperre.

```
def get_legal_hold(s3_client, bucket: str, key: str) -> None:
    """
    Get the legal hold status of a specific file in a bucket.

    Args:
        s3_client: Boto3 S3 client.
        bucket: The name of the bucket containing the file.
        key: The key of the file to get the legal hold status of.
    """
    print()
    logger.info("Getting legal hold status of file [%s] in bucket [%s]", key,
                bucket)
    try:
        response = s3_client.get_object_legal_hold(Bucket=bucket, Key=key)
        legal_hold_status = response["LegalHold"]["Status"]
        logger.debug(
```



```
        "Legal hold status of file [%s] in bucket [%s] is [%s]",
        key,
        bucket,
        legal_hold_status,
    )
except Exception as e:
    logger.error(
        "Failed to get legal hold status of file [%s] in bucket [%s]: %s",
        key,
        bucket,
        e,
    )
```

- API-Einheiten finden Sie [GetObjectLegalHold](#) in AWS SDK für Python (Boto3) API - Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetObjectLockConfiguration** mit einem AWS SDK oder CLI.


Die folgenden Codebeispiele zeigen die Verwendung von `GetObjectLockConfiguration`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Amazon S3 S3-Objekte sperren](#)

.NET

AWS SDK for .NET

 Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Get the object lock configuration details for an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket to get details.</param>
/// <returns>The bucket's object lock configuration details.</returns>
public async Task<ObjectLockConfiguration>
GetBucketObjectLockConfiguration(string bucketName)
{
    try
    {
        var request = new GetObjectLockConfigurationRequest()
        {
            BucketName = bucketName
        };

        var response = await
        _amazonS3.GetObjectLockConfigurationAsync(request);
        Console.WriteLine($"\\tBucket object lock config for {bucketName} in
{bucketName}: " +
            $"\\n\\tEnabled:
{response.ObjectLockConfiguration.ObjectLockEnabled}" +
            $"\\n\\tRule:
{response.ObjectLockConfiguration.Rule?.DefaultRetention}");

        return response.ObjectLockConfiguration;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tUnable to fetch object lock config:
'{ex.Message}'");
        return new ObjectLockConfiguration();
    }
}
```

- API-Einheiten finden Sie [GetObjectLockConfiguration](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um eine Objektsperrenkonfiguration für einen Bucket abzurufen

Im folgenden `get-object-lock-configuration` Beispiel wird die Objektsperrenkonfiguration für den angegebenen Bucket abgerufen.

```
aws s3api get-object-lock-configuration \  
  --bucket my-bucket-with-object-lock
```

Ausgabe:

```
{  
  "ObjectLockConfiguration": {  
    "ObjectLockEnabled": "Enabled",  
    "Rule": {  
      "DefaultRetention": {  
        "Mode": "COMPLIANCE",  
        "Days": 50  
      }  
    }  
  }  
}
```

- API-Einheiten finden Sie unter [GetObjectLockConfiguration](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.
```

```
type S3Actions struct {
    S3Client    *s3.Client
    S3Manager   *manager.Uploader
}

// GetObjectLockConfiguration retrieves the object lock configuration for an S3
// bucket.
func (actor S3Actions) GetObjectLockConfiguration(ctx context.Context, bucket
string) (*types.ObjectLockConfiguration, error) {
    var lockConfig *types.ObjectLockConfiguration
    input := &s3.GetObjectLockConfigurationInput{
        Bucket: aws.String(bucket),
    }

    output, err := actor.S3Client.GetObjectLockConfiguration(ctx, input)
    if err != nil {
        var noBucket *types.NoSuchBucket
        var apiErr *smithy.GenericAPIError
        if errors.As(err, &noBucket) {
            log.Printf("Bucket %s does not exist.\n", bucket)
            err = noBucket
        } else if errors.As(err, &apiErr) && apiErr.ErrorCode() ==
"ObjectLockConfigurationNotFoundError" {
            log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
            err = nil
        }
    } else {
        lockConfig = output.ObjectLockConfiguration
    }

    return lockConfig, err
}
```

- APIEinzelheiten finden Sie [GetObjectLockConfiguration](#) unter AWS SDK for Go APIReferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// Get the object lock configuration details for an S3 bucket.
public void getBucketObjectLockConfiguration(String bucketName) {
    GetObjectLockConfigurationRequest objectLockConfigurationRequest =
GetObjectLockConfigurationRequest.builder()
        .bucket(bucketName)
        .build();

    GetObjectLockConfigurationResponse response =
getClient().getObjectLockConfiguration(objectLockConfigurationRequest);
    System.out.println("Bucket object lock config for "+bucketName+": ");
    System.out.println("\tEnabled:
"+response.getObjectLockConfiguration().objectLockEnabled());
    System.out.println("\tRule: "+
response.getObjectLockConfiguration().rule().defaultRetention());
}
```

- API Einzelheiten finden Sie [GetObjectLockConfiguration](#) unter AWS SDK for Java 2.x API Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import { fileURLToPath } from "url";
import {
  GetObjectLockConfigurationCommand,
  S3Client,
} from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 */
export const main = async (client, bucketName) => {
  const command = new GetObjectLockConfigurationCommand({
    Bucket: bucketName,
    // Optionally, you can provide additional parameters
    // ExpectedBucketOwner: "ACCOUNT_ID",
  });

  try {
    const { ObjectLockConfiguration } = await client.send(command);
    console.log(`Object Lock Configuration: ${ObjectLockConfiguration}`);
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME");
}
```

- API Einzelheiten finden Sie [GetObjectLockConfiguration](#) unter AWS SDK for JavaScript API Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt den Wert 'Enabled' zurück, wenn die Objektsperrekonfiguration für den angegebenen S3-Bucket aktiviert ist.

```
Get-S3ObjectLockConfiguration -BucketName 's3buckettesting' -Select  
ObjectLockConfiguration.ObjectLockEnabled
```

Ausgabe:

```
Value  
-----  
Enabled
```

- API-Einheiten finden Sie [GetObjectLockConfiguration](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Ruft die Objektsperren-Konfiguration ab.

```
def is_object_lock_enabled(s3_client, bucket: str) -> bool:  
    """  
    Check if object lock is enabled for a bucket.  
  
    Args:  
        s3_client: Boto3 S3 client.  
        bucket: The name of the bucket to check.  
  
    Returns:  
        True if object lock is enabled, False otherwise.  
    """  
    try:  
        response = s3_client.get_object_lock_configuration(Bucket=bucket)  
        return (  
            "ObjectLockConfiguration" in response  
            and response["ObjectLockConfiguration"]["ObjectLockEnabled"] ==  
            "Enabled"
```

```
)  
except s3_client.exceptions.ClientError as e:  
    if e.response["Error"]["Code"] == "ObjectLockConfigurationNotFoundError":  
        return False  
    else:  
        raise
```

- API-Einheiten finden Sie [GetObjectLockConfiguration](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetObjectRetention** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetObjectRetention`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Amazon S3 S3-Objekte sperren](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>  
/// Get the retention period for an S3 object.  
/// </summary>
```



```

/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object retention details.</returns>
public async Task<ObjectLockRetention> GetObjectRetention(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectRetentionRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };

        var response = await _amazonS3.GetObjectRetentionAsync(request);
        Console.WriteLine($"{\tObject retention for {objectKey} in
{bucketName}: " +
            $"{\n\t{response.Retention.Mode} until
{response.Retention.RetainUntilDate:d}."");
        return response.Retention;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"{\tUnable to fetch object lock retention:
'{ex.Message}'");
        return new ObjectLockRetention();
    }
}

```

- API-Einheiten finden Sie [GetObjectRetention](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um die Objektaufbewahrungskonfiguration für ein Objekt abzurufen

Im folgenden `get-object-retention` Beispiel wird die Objektaufbewahrungskonfiguration für das angegebene Objekt abgerufen.

```

aws s3api get-object-retention \
    --bucket my-bucket-with-object-lock \

```

```
--key doc1.rtf
```


Ausgabe:

```
{
  "Retention": {
    "Mode": "GOVERNANCE",
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"
  }
}
```

- API-Einheiten finden Sie unter [GetObjectRetention](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
  S3Client *s3.Client
  S3Manager *manager.Uploader
}

// GetObjectRetention retrieves the object retention configuration for an S3
// object.
func (actor S3Actions) GetObjectRetention(ctx context.Context, bucket string, key
string) (*types.ObjectLockRetention, error) {
  var retention *types.ObjectLockRetention
  input := &s3.GetObjectRetentionInput{
    Bucket: aws.String(bucket),
    Key:   aws.String(key),
  }
}
```

```
output, err := actor.S3Client.GetObjectRetention(ctx, input)
if err != nil {
    var noKey *types.NoSuchKey
    var apiErr *smithy.GenericAPIError
    if errors.As(err, &noKey) {
        log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
        err = noKey
    } else if errors.As(err, &apiErr) {
        switch apiErr.ErrorCode() {
        case "NoSuchObjectLockConfiguration":
            err = nil
        case "InvalidRequest":
            log.Printf("Bucket %s does not have locking enabled.", bucket)
            err = nil
        }
    }
} else {
    retention = output.Retention
}

return retention, err
}
```

- API-Einheiten finden Sie [GetObjectRetention](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// Get the retention period for an S3 object.
public ObjectLockRetention getObjectRetention(String bucketName, String key){
    try {
```

```

        GetObjectRetentionRequest retentionRequest =
GetObjectRetentionRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();

        GetObjectRetentionResponse response =
getClient().getObjectRetention(retentionRequest);
        System.out.println("tObject retention for "+key +"
in "+ bucketName +": " + response.retention().mode() +" until "+
response.retention().retainUntilDate() +".");
        return response.retention();

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        return null;
    }
}

```

- APIEinzelheiten finden Sie [GetObjectRetention](#) unter AWS SDK for Java 2.x APIReferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

import { fileURLToPath } from "url";
import { GetObjectRetentionCommand, S3Client } from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
    const command = new GetObjectRetentionCommand({

```

```
    Bucket: bucketName,
    Key: objectKey,
    // Optionally, you can provide additional parameters
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    // VersionId: "OBJECT_VERSION_ID",
  });

  try {
    const { Retention } = await client.send(command);
    console.log(`Object Retention Settings: ${Retention.Status}`);
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME", "OBJECT_KEY");
}
```

- API-Einheiten finden Sie [GetObjectRetention](#) unter AWS SDK for JavaScript API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl gibt den Modus und das Datum zurück, bis das Objekt beibehalten werden würde.

```
Get-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt'
```

- API-Einheiten finden Sie [GetObjectRetention](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetObjectTagging** mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen die Verwendung `GetObjectTagging`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erste Schritte mit Tags](#)

CLI

AWS CLI

Um die an ein Objekt angehängten Tags abzurufen

Im folgenden `get-object-tagging` Beispiel werden die Werte für den angegebenen Schlüssel aus dem angegebenen Objekt abgerufen.

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

Ausgabe:

```
{  
  "TagSet": [  
    {  
      "Value": "confidential",  
      "Key": "designation"  
    }  
  ]  
}
```

Im folgenden `get-object-tagging` Beispiel wird versucht, die Tagsätze des Objekts abzurufen `doc2.rtf`, das keine Tags hat.

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc2.rtf
```

Ausgabe:

```
{
  "TagSet": []
}
```

Im folgenden `get-object-tagging` Beispiel werden die Tagsätze des Objekts `doc3.rtf`, das über mehrere Tags verfügt.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc3.rtf
```

Ausgabe:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    },
    {
      "Value": "finance",
      "Key": "department"
    },
    {
      "Value": "payroll",
      "Key": "team"
    }
  ]
}
```

- API-Einheiten finden Sie unter [GetObjectTagging](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Das Beispiel gibt die Tags zurück, die dem Objekt zugeordnet sind, das im angegebenen S3-Bucket vorhanden ist.

```
Get-S3ObjectTagSet -Key 'testfile.txt' -BucketName 'testbucket123'
```

Ausgabe:

```
Key Value
--- ----
test value
```

- API-Einheiten finden Sie [GetObjectTagging](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetPublicAccessBlock** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `GetPublicAccessBlock`.

CLI

AWS CLI

Um die Konfiguration für den öffentlichen Zugriff zu blockieren für einen Bucket festzulegen oder zu ändern

Das folgende `get-public-access-block` Beispiel zeigt die Konfiguration für den blockierten öffentlichen Zugriff für den angegebenen Bucket.

```
aws s3api get-public-access-block \
  --bucket my-bucket
```

Ausgabe:

```
{
  "PublicAccessBlockConfiguration": {
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "BlockPublicAcls": true,
```



```

    "RestrictPublicBuckets": true
  }
}

```

- API-Einheiten finden Sie [GetPublicAccessBlock](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl gibt die Blockkonfiguration für den öffentlichen Zugriff des angegebenen S3-Buckets zurück.

```
Get-S3PublicAccessBlock -BucketName 's3testbucket'
```

- API-Einheiten finden Sie [GetPublicAccessBlock](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **HeadBucket** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `HeadBucket`.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#####
# function bucket_exists
```

```

#
# This function checks to see if the specified bucket already exists.
#
# Parameters:
#     $1 - The name of the bucket to check.
#
# Returns:
#     0 - If the bucket already exists.
#     1 - If the bucket doesn't exist.
#####
function bucket_exists() {
    local bucket_name
    bucket_name=$1

    # Check whether the bucket already exists.
    # We suppress all output - we're interested only in the return code.

    if aws s3api head-bucket \
        --bucket "$bucket_name" \
        >/dev/null 2>&1; then
        return 0 # 0 in Bash script means true.
    else
        return 1 # 1 in Bash script means false.
    fi
}

```

- API-Einheiten finden Sie [HeadBucket](#) unter AWS CLI Befehlsreferenz.

CLI

AWS CLI

Der folgende Befehl überprüft den Zugriff auf einen Bucket mit dem Namen `my-bucket`:

```
aws s3api head-bucket --bucket my-bucket
```


Wenn der Bucket existiert und Sie Zugriff darauf haben, wird keine Ausgabe zurückgegeben. Andernfalls wird eine Fehlermeldung angezeigt. Beispielsweise:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- API-Einheiten finden Sie [HeadBucket](#) in AWS CLI Befehlsreferenz.

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// BucketExists checks whether a bucket exists in the current account.
func (basics BucketBasics) BucketExists(bucketName string) (bool, error) {
    _, err := basics.S3Client.HeadBucket(context.TODO(), &s3.HeadBucketInput{
        Bucket: aws.String(bucketName),
    })
    exists := true
    if err != nil {
        var apiError smithy.APIError
        if errors.As(err, &apiError) {
            switch apiError.(type) {
            case *types.NotFound:
                log.Printf("Bucket %v is available.\n", bucketName)
                exists = false
                err = nil
            default:
                log.Printf("Either you don't have access to bucket %v or another error
                occurred. "+
```

```
    "Here's what happened: %v\n", bucketName, err)
  }
} else {
  log.Printf("Bucket %v exists and you already own it.", bucketName)
}

return exists, err
}
```

- API-Einheiten finden Sie [HeadBucket](#) unter AWS SDK for Go API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def exists(self):
        """
        Determine whether the bucket exists and you have access to it.

        :return: True when the bucket exists; otherwise, False.
        """
```

```
"""
try:
    self.bucket.meta.client.head_bucket(Bucket=self.bucket.name)
    logger.info("Bucket %s exists.", self.bucket.name)
    exists = True
except ClientError:
    logger.warning(
        "Bucket %s doesn't exist or you don't have access to it.",
        self.bucket.name,
    )
    exists = False
return exists
```

- API-Einheiten finden Sie [HeadBucket](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **HeadObject** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `HeadObject`.

CLI

AWS CLI

Der folgende Befehl ruft Metadaten für ein Objekt in einem Bucket mit dem Namen `my-bucket` ab:

```
aws s3api head-object --bucket my-bucket --key index.html
```

Ausgabe:

```
{
  "AcceptRanges": "bytes",
  "ContentType": "text/html",
  "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
  "ContentLength": 77,
```

```
"VersionId": "null",
"ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
"Metadata": {}
}
```

- API-Einheiten finden Sie unter [HeadObject](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Bestimmen Sie den Inhaltstyp eines Objekts.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class GetObjectContentType {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName> <keyName>>

                Where:
```

```
        bucketName - The Amazon S3 bucket name.\s
        keyName - The key name.\s
        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String bucketName = args[0];
    String keyName = args[1];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    getContentType(s3, bucketName, keyName);
    s3.close();
}

public static void getContentType(S3Client s3, String bucketName, String
keyName) {
    try {
        HeadObjectRequest objectRequest = HeadObjectRequest.builder()
            .key(keyName)
            .bucket(bucketName)
            .build();

        HeadObjectResponse objectHead = s3.headObject(objectRequest);
        String type = objectHead.contentType();
        System.out.println("The object content type is " + type);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

Rufen Sie den Wiederherstellungsstatus eines Objekts ab.

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

public class GetObjectRestoreStatus {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <keyName>\s

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - A key name that represents the object.\s
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        checkStatus(s3, bucketName, keyName);
        s3.close();
    }

    public static void checkStatus(S3Client s3, String bucketName, String
keyName) {
        try {
            HeadObjectRequest headObjectRequest = HeadObjectRequest.builder()
                .bucket(bucketName)
                .key(keyName)
                .build();

            HeadObjectResponse response = s3.headObject(headObjectRequest);
            System.out.println("The Amazon S3 object restoration status is " +
response.restore());
        }
    }
}
```



```
    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- API-Einheiten finden Sie [HeadObject](#) unter AWS SDK for Java 2.x API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectExistsWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Checks whether the object exists.
  #
  # @return [Boolean] True if the object exists; otherwise false.
  def exists?
    @object.exists?
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't check existence of object
    #{@object.bucket.name}:#{@object.key}. Here's why: #{e.message}"
    false
  end
end
```

```
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectExistsWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  exists = wrapper.exists?

  puts "Object #{object_key} #{exists ? 'does' : 'does not'} exist."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- API-Einheiten finden Sie [HeadObject](#) unter AWS SDK for Ruby API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListBucketAnalyticsConfigurations** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `ListBucketAnalyticsConfigurations`.

CLI

AWS CLI

Um eine Liste von Analytics-Konfigurationen für einen Bucket abzurufen

Im Folgenden wird eine Liste der Analytics-Konfigurationen für den angegebenen Bucket `list-bucket-analytics-configurations` abgerufen.

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

Ausgabe:

```
{
```

```
"AnalyticsConfigurationList": [  
  {  
    "StorageClassAnalysis": {},  
    "Id": "1"  
  }  
],  
"IsTruncated": false  
}
```

- API-Einheiten finden Sie unter [ListBucketAnalyticsConfigurations](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die ersten 100 Analytics-Konfigurationen des angegebenen S3-Buckets zurück.

```
Get-S3BucketAnalyticsConfigurationList -BucketName 's3casetestbucket'
```

- API-Einheiten finden Sie [ListBucketAnalyticsConfigurations](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListBucketInventoryConfigurations** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `ListBucketInventoryConfigurations`.

CLI

AWS CLI

Um eine Liste von Inventarkonfigurationen für einen Bucket abzurufen

Das folgende `list-bucket-inventory-configurations` Beispiel listet die Inventarkonfigurationen für den angegebenen Bucket auf.

```
aws s3api list-bucket-inventory-configurations \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "InventoryConfigurationList": [  
    {  
      "IsEnabled": true,  
      "Destination": {  
        "S3BucketDestination": {  
          "Format": "ORC",  
          "Bucket": "arn:aws:s3:::my-bucket",  
          "AccountId": "123456789012"  
        }  
      },  
      "IncludedObjectVersions": "Current",  
      "Id": "1",  
      "Schedule": {  
        "Frequency": "Weekly"  
      }  
    },  
    {  
      "IsEnabled": true,  
      "Destination": {  
        "S3BucketDestination": {  
          "Format": "CSV",  
          "Bucket": "arn:aws:s3:::my-bucket",  
          "AccountId": "123456789012"  
        }  
      },  
      "IncludedObjectVersions": "Current",  
      "Id": "2",  
      "Schedule": {  
        "Frequency": "Daily"  
      }  
    }  
  ],  
  "IsTruncated": false  
}
```

- API-Einheiten finden Sie [ListBucketInventoryConfigurations](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt die ersten 100 Inventarkonfigurationen des angegebenen S3-Buckets zurück.

```
Get-S3BucketInventoryConfigurationList -BucketName 's3testbucket'
```

- API-Einheiten finden Sie [ListBucketInventoryConfigurations](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListBuckets** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung `ListBuckets`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
namespace ListBucketsExample
{
    using System;
    using System.Collections.Generic;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;
```

```
/// <summary>
/// This example uses the AWS SDK for .NET to list the Amazon Simple Storage
/// Service (Amazon S3) buckets belonging to the default account.
/// </summary>
public class ListBuckets
{
    private static IAmazonS3 _s3Client;

    /// <summary>
    /// Get a list of the buckets owned by the default user.
    /// </summary>
    /// <param name="client">An initialized Amazon S3 client object.</param>
    /// <returns>The response from the ListingBuckets call that contains a
    /// list of the buckets owned by the default user.</returns>
    public static async Task<ListBucketsResponse> GetBuckets(IAmazonS3
client)
    {
        return await client.ListBucketsAsync();
    }

    /// <summary>
    /// This method lists the name and creation date for the buckets in
    /// the passed List of S3 buckets.
    /// </summary>
    /// <param name="bucketList">A List of S3 bucket objects.</param>
    public static void DisplayBucketList(List<S3Bucket> bucketList)
    {
        bucketList
            .ForEach(b => Console.WriteLine($"Bucket name: {b.BucketName},
created on: {b.CreationDate}"));
    }

    public static async Task Main()
    {
        // The client uses the AWS Region of the default user.
        // If the Region where the buckets were created is different,
        // pass the Region to the client constructor. For example:
        // _s3Client = new AmazonS3Client(RegionEndpoint.USEast1);
        _s3Client = new AmazonS3Client();
        var response = await GetBuckets(_s3Client);
        DisplayBucketList(response.Buckets);
    }
}
```

```
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for .NET API-Referenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::listBuckets(const Aws::S3::S3ClientConfiguration &clientConfig)
{
    Aws::S3::S3Client client(clientConfig);

    auto outcome = client.ListBuckets();

    bool result = true;
    if (!outcome.IsSuccess()) {
        std::cerr << "Failed with error: " << outcome.GetError() << std::endl;
        result = false;
    } else {
        std::cout << "Found " << outcome.GetResult().GetBuckets().size() << "
buckets\n";
        for (auto &&b: outcome.GetResult().GetBuckets()) {
            std::cout << b.GetName() << std::endl;
        }
    }

    return result;
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl verwendet den `list-buckets` Befehl, um die Namen all Ihrer Amazon S3 S3-Buckets (in allen Regionen) anzuzeigen:

```
aws s3api list-buckets --query "Buckets[].Name"
```

Die Abfrageoption filtert die Ausgabe von `list-buckets` bis auf die Bucket-Namen.

Weitere Informationen zu Buckets finden Sie unter [Working with Amazon S3 Buckets](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie unter [ListBuckets](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu [GitHub](#). Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// ListBuckets lists the buckets in the current account.
func (basics BucketBasics) ListBuckets() ([]types.Bucket, error) {
```



```
result, err := basics.S3Client.ListBuckets(context.TODO(),
&s3.ListBucketsInput{})
var buckets []types.Bucket
if err != nil {
    log.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
} else {
    buckets = result.Buckets
}
return buckets, err
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ListBuckets {
    public static void main(String[] args) {
```

```
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    listAllBuckets(s3);

}

public static void listAllBuckets(S3Client s3) {
    ListBucketsResponse response = s3.listBuckets();
    List<Bucket> bucketList = response.buckets();
    for (Bucket bucket: bucketList) {
        System.out.println("Bucket name "+bucket.name());
    }
}
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Listen Sie die Buckets auf.

```
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
    const command = new ListBucketsCommand({});

    try {
        const { Owner, Buckets } = await client.send(command);
        console.log(
```

```
    `${Owner.DisplayName} owns ${Buckets.length} bucket${  
      Buckets.length === 1 ? "" : "s"  
    }:` ,  
  );  
  console.log(`${Buckets.map((b) => ` • ${b.Name}`).join("\n")}`);  
} catch (err) {  
  console.error(err);  
}  
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [ListBuckets](#) in AWS SDK for JavaScript API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl gibt alle S3-Buckets zurück.

```
Get-S3Bucket
```

Beispiel 2: Dieser Befehl gibt einen Bucket mit dem Namen „test-files“ zurück

```
Get-S3Bucket -BucketName test-files
```

- API-Einheiten finden Sie in [ListBuckets](#) AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
```

```
"""Encapsulates S3 bucket actions."""

def __init__(self, bucket):
    """
    :param bucket: A Boto3 Bucket resource. This is a high-level resource in
    Boto3
                   that wraps bucket actions in a class-like structure.
    """
    self.bucket = bucket
    self.name = bucket.name

    @staticmethod
    def list(s3_resource):
        """
        Get the buckets in all Regions for the current account.

        :param s3_resource: A Boto3 S3 resource. This is a high-level resource in
        Boto3
                           that contains collections and factory methods to
        create
                           other high-level S3 sub-resources.
        :return: The list of buckets.
        """
        try:
            buckets = list(s3_resource.buckets.all())
            logger.info("Got buckets: %s.", buckets)
        except ClientError:
            logger.exception("Couldn't get buckets.")
            raise
        else:
            return buckets
```

- API Einzelheiten finden Sie [ListBuckets](#) unter AWS SDK für Python (Boto3) API -Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 resource actions.
class BucketListWrapper
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Lists buckets for the current account.
  #
  # @param count [Integer] The maximum number of buckets to list.
  def list_buckets(count)
    puts "Found these buckets:"
    @s3_resource.buckets.each do |bucket|
      puts "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list buckets. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  wrapper = BucketListWrapper.new(Aws::S3::Resource.new)
  wrapper.list_buckets(25)
end
```

```
end

run_demo if $PROGRAM_NAME == __FILE__
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
async fn show_buckets(
    strict: bool,
    client: &Client,
    region: BucketLocationConstraint,
) -> Result<(), S3ExampleError> {
    let mut buckets = client.list_buckets().into_paginator().send();

    let mut num_buckets = 0;
    let mut in_region = 0;

    while let Some(Ok(output)) = buckets.next().await {
        for bucket in output.buckets() {
            num_buckets += 1;
            if strict {
                let r = client
                    .get_bucket_location()
                    .bucket(bucket.name().unwrap_or_default())
                    .send()
                    .await?;

                if r.location_constraint() == Some(&region) {
                    println!("{}", bucket.name().unwrap_or_default());
                    in_region += 1;
                }
            }
        }
    } else {
```

```
        println!("{}", bucket.name().unwrap_or_default());
    }
}

println!();
if strict {
    println!(
        "Found {} buckets in the {} region out of a total of {} buckets.",
        in_region, region, num_buckets
    );
} else {
    println!("Found {} buckets in all regions.", num_buckets);
}

Ok(())
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK als API Referenz auf Rust.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3

/// Return an array containing information about every available bucket.
///
/// - Returns: An array of ``S3ClientTypes.Bucket`` objects describing
/// each bucket.
public func getAllBuckets() async throws -> [S3ClientTypes.Bucket] {
    return try await client.listBuckets(input: ListBucketsInput())
}
```

- API-Einheiten finden Sie [ListBuckets](#) unter AWS SDK als API-Swift-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListMultipartUploads** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen die Verwendung von `ListMultipartUploads`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Löschen Sie unvollständige mehrteilige Uploads](#)

CLI

AWS CLI

Der folgende Befehl listet alle aktiven mehrteiligen Uploads für einen Bucket mit dem Namen `my-bucket` auf:

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Ausgabe:

```
{
  "Uploads": [
    {
      "Initiator": {
        "DisplayName": "username",
        "ID": "arn:aws:iam::0123456789012:user/username"
      },
      "Initiated": "2015-06-02T18:01:30.000Z",
      "UploadId":
"dfRtDYU0WwCCcH43C3WfbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3",
      "StorageClass": "STANDARD",
      "Key": "multipart/01",
      "Owner": {
        "DisplayName": "aws-account-name",

```



```
        "ID":
        "100719349fc3b6dcd7c820a124bf7aec408092c3d7b51b38494939801fc248b"
    }
  ],
  "CommonPrefixes": []
}
```

Bei laufenden mehrteiligen Uploads fallen Speicherkosten in Amazon S3 an. Schließen Sie einen aktiven mehrteiligen Upload ab oder brechen Sie ihn ab, um seine Teile aus Ihrem Konto zu entfernen.

- API-Einheiten finden Sie unter [ListMultipartUploads](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListMultipartUploadsRequest;
import software.amazon.awssdk.services.s3.model.ListMultipartUploadsResponse;
import software.amazon.awssdk.services.s3.model.MultipartUpload;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
```

```
public class ListMultipartUploads {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName>\s

            Where:
                bucketName - The name of the Amazon S3 bucket where an in-
progress multipart upload is occurring.
            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();
        listUploads(s3, bucketName);
        s3.close();
    }

    public static void listUploads(S3Client s3, String bucketName) {
        try {
            ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
                .bucket(bucketName)
                .build();

            ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
            List<MultipartUpload> uploads = response.uploads();
            for (MultipartUpload upload : uploads) {
                System.out.println("Upload in progress: Key = \"\" + upload.key()
+ "\", id = \"\" + upload.uploadId());
            }

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

```
    }  
  }  
}
```

- API-Einheiten finden Sie [ListMultipartUploads](#) unter AWS SDK for Java 2.x API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListObjectVersions** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `ListObjectVersions`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Mit versionierten Objekten arbeiten](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;  
using System.Threading.Tasks;  
using Amazon.S3;  
using Amazon.S3.Model;  
  
/// <summary>  
/// This example lists the versions of the objects in a version enabled  
/// Amazon Simple Storage Service (Amazon S3) bucket.  
/// </summary>  
public class ListObjectVersions
```

```
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";

        // If the AWS Region where your bucket is defined is different from
        // the AWS Region where the Amazon S3 bucket is defined, pass the
constant
        // for the AWS Region to the client constructor like this:
        //     var client = new AmazonS3Client(RegionEndpoint.USWest2);
        IAmazonS3 client = new AmazonS3Client();
        await GetObjectListWithAllVersionsAsync(client, bucketName);
    }

    /// <summary>
    /// This method lists all versions of the objects within an Amazon S3
    /// version enabled bucket.
    /// </summary>
    /// <param name="client">The initialized client object used to call
    /// ListVersionsAsync.</param>
    /// <param name="bucketName">The name of the version enabled Amazon S3
bucket
    /// for which you want to list the versions of the contained objects.</
param>
    public static async Task GetObjectListWithAllVersionsAsync(IAmazonS3
client, string bucketName)
    {
        try
        {
            // When you instantiate the ListVersionRequest, you can
            // optionally specify a key name prefix in the request
            // if you want a list of object versions of a specific object.

            // For this example we set a small limit in MaxKeys to return
            // a small list of versions.
            ListVersionsRequest request = new ListVersionsRequest()
            {
                BucketName = bucketName,
                MaxKeys = 2,
            };

            do
            {
```

```
        ListVersionsResponse response = await
client.ListVersionsAsync(request);

        // Process response.
        foreach (S3ObjectVersion entry in response.Versions)
        {
            Console.WriteLine($"key: {entry.Key} size:
{entry.Size}");
        }

        // If response is truncated, set the marker to get the next
        // set of keys.
        if (response.IsTruncated)
        {
            request.KeyMarker = response.NextKeyMarker;
            request.VersionIdMarker = response.NextVersionIdMarker;
        }
        else
        {
            request = null;
        }
    }
    while (request != null);
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error: '{ex.Message}'");
}
}
```

- API-Einheiten finden Sie [ListObjectVersions](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Der folgende Befehl ruft Versionsinformationen für ein Objekt in einem Bucket mit dem Namen `my-bucket` ab:

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Ausgabe:

```
{
  "DeleteMarkers": [
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": true,
      "VersionId": "B2VsEK5saUNNHKc0AJj7hIE86RozToyq",
      "Key": "index.html",
      "LastModified": "2015-11-10T00:57:03.000Z"
    },
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": false,
      "VersionId": ".FLQEZscLIcfxSq.jsFJ.szUkmng2Yw6",
      "Key": "index.html",
      "LastModified": "2015-11-09T23:32:20.000Z"
    }
  ],
  "Versions": [
    {
      "LastModified": "2015-11-10T00:20:11.000Z",
      "VersionId": "Rb_l2T8UHDkFEwCgJjhlGPOZC0qJ.vpD",
      "ETag": "\"0622528de826c0df5db1258a23b80be5\"",
      "StorageClass": "STANDARD",
      "Key": "index.html",
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": false,
      "Size": 38
    }
  ]
}
```

```

    },
    {
      "LastModified": "2015-11-09T23:26:41.000Z",
      "VersionId": "rasWWGpgk9E4s0LyTJgusGeRQKLVIAff",
      "ETag": "\"06225825b8028de826c0df5db1a23be5\"",
      "StorageClass": "STANDARD",
      "Key": "index.html",
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": false,
      "Size": 38
    },
  ],
  {
    "LastModified": "2015-11-09T22:50:50.000Z",
    "VersionId": "null",
    "ETag": "\"d1f45267a863c8392e07d24dd592f1b9\"",
    "StorageClass": "STANDARD",
    "Key": "index.html",
    "Owner": {
      "DisplayName": "my-username",
      "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
    },
    "IsLatest": false,
    "Size": 533823
  }
]
}

```

- API-Einheiten finden Sie unter [ListObjectVersions](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
    S3Client    *s3.Client
    S3Manager   *manager.Uploader
}

// ListObjectVersions lists all versions of all objects in a bucket.
func (actor S3Actions) ListObjectVersions(ctx context.Context, bucket string)
([]types.ObjectVersion, error) {
    var err error
    var output *s3.ListObjectVersionsOutput
    var versions []types.ObjectVersion
    input := &s3.ListObjectVersionsInput{Bucket: aws.String(bucket)}
    versionPaginator := s3.NewListObjectVersionsPaginator(actor.S3Client, input)
    for versionPaginator.HasMorePages() {
        output, err = versionPaginator.NextPage(ctx)
        if err != nil {
            var noBucket *types.NoSuchBucket
            if errors.As(err, &noBucket) {
                log.Printf("Bucket %s does not exist.\n", bucket)
                err = noBucket
            }
            break
        } else {
            versions = append(versions, output.Versions...)
        }
    }
    return versions, err
}
```

- API-Einheiten finden Sie [ListObjectVersions](#) unter AWS SDK for Go API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
async fn show_versions(client: &Client, bucket: &str) -> Result<(), Error> {
    let resp = client.list_object_versions().bucket(bucket).send().await?;

    for version in resp.versions() {
        println!("{}", version.key().unwrap_or_default());
        println!(" version ID: {}", version.version_id().unwrap_or_default());
        println!();
    }

    Ok(())
}
```

- API Einzelheiten finden Sie [ListObjectVersions](#) unter AWS SDK als API Referenz auf Rust.

Für eine vollständige Liste von AWS SDK Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **ListObjects** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `ListObjects`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Eine Webseite erstellen, die Amazon-S3-Objekte auflistet](#)

CLI

AWS CLI

Im folgenden Beispiel wird der `list-objects` Befehl verwendet, um die Namen aller Objekte im angegebenen Bucket anzuzeigen:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

Im Beispiel wird das `--query` Argument verwendet, um die Ausgabe von `list-objects` nach Schlüsselwert und Größe für jedes Objekt zu filtern

Weitere Informationen zu Objekten finden Sie unter [Working with Amazon S3 Objects](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie [ListObjects](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl ruft die Informationen über alle Elemente im Bucket „test-files“ ab.

```
Get-S3Object -BucketName test-files
```

Beispiel 2: Dieser Befehl ruft die Informationen über das Objekt "sample.txt" aus dem Bucket „test-files“ ab.

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Beispiel 3: Dieser Befehl ruft die Informationen über alle Elemente mit dem Präfix „sample“ aus dem Bucket „test-files“ ab.

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- API-Einheiten finden Sie unter [ListObjects](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListObjectsV2** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `ListObjectsV2`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erlernen Sie die Grundlagen](#)
- [Löscht alle Objekte in einem Bucket](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Shows how to list the objects in an Amazon S3 bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The name of the bucket for which to list
/// the contents.</param>
/// <returns>A boolean value indicating the success or failure of the
/// copy operation.</returns>
public static async Task<bool> ListBucketContentsAsync(IAmazonS3 client,
string bucketName)
{
    try
    {
        var request = new ListObjectsV2Request
        {
```

```
        BucketName = bucketName,
        MaxKeys = 5,
    };

    Console.WriteLine("-----");
    Console.WriteLine($"Listing the contents of {bucketName}:");
    Console.WriteLine("-----");

    ListObjectsV2Response response;

    do
    {
        response = await client.ListObjectsV2Async(request);

        response.S3Objects
            .ForEach(obj => Console.WriteLine($"{obj.Key,-35}
{obj.LastModified.ToShortDateString(),10}{obj.Size,10}"));

        // If the response is truncated, set the request
        ContinuationToken
            // from the NextContinuationToken property of the response.
            request.ContinuationToken = response.NextContinuationToken;
    }
    while (response.IsTruncated);

    return true;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error encountered on server.
Message: '{ex.Message}' getting list of objects.");
    return false;
}
}
```

Listen Sie Objekte mit einem Paginator auf.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
```

```
/// <summary>
/// The following example lists objects in an Amazon Simple Storage
/// Service (Amazon S3) bucket.
/// </summary>
public class ListObjectsPaginator
{
    private const string BucketName = "amzn-s3-demo-bucket";

    public static async Task Main()
    {
        IAmazonS3 s3Client = new AmazonS3Client();

        Console.WriteLine($"Listing the objects contained in {BucketName}:
\n");
        await ListingObjectsAsync(s3Client, BucketName);
    }

    /// <summary>
    /// This method uses a paginator to retrieve the list of objects in an
    /// an Amazon S3 bucket.
    /// </summary>
    /// <param name="client">An Amazon S3 client object.</param>
    /// <param name="bucketName">The name of the S3 bucket whose objects
    /// you want to list.</param>
    public static async Task ListingObjectsAsync(IAmazonS3 client, string
bucketName)
    {
        var listObjectsV2Paginator = client.Paginators.ListObjectsV2(new
ListObjectsV2Request
        {
            BucketName = bucketName,
        });

        await foreach (var response in listObjectsV2Paginator.Responses)
        {
            Console.WriteLine($"HttpStatusCode: {response.HttpStatusCode}");
            Console.WriteLine($"Number of Keys: {response.KeyCount}");
            foreach (var entry in response.S3Objects)
            {
                Console.WriteLine($"Key = {entry.Key} Size = {entry.Size}");
            }
        }
    }
}
```

```
}

```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function list_items_in_bucket() {
```

```

local bucket_name=$1
local response

response=$(aws s3api list-objects \
  --bucket "$bucket_name" \
  --output text \
  --query 'Contents[].{Key: Key, Size: Size}')

# shellcheck disable=SC2181
if [[ ${?} -eq 0 ]]; then
  echo "$response"
else
  errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
  return 1
fi
}

```

- API-Einzelheiten finden Sie unter [ListObjectsV2](#) in AWS CLI Befehlsreferenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

bool AwsDoc::S3::listObjects(const Aws::String &bucketName,
                             Aws::Vector<Aws::String> &keysResult,
                             const Aws::S3::S3ClientConfiguration &clientConfig)
{
  Aws::S3::S3Client s3Client(clientConfig);

  Aws::S3::Model::ListObjectsV2Request request;
  request.WithBucket(bucketName);

  Aws::String continuationToken; // Used for pagination.
  Aws::Vector<Aws::S3::Model::Object> allObjects;

```

```
do {
    if (!continuationToken.empty()) {
        request.SetContinuationToken(continuationToken);
    }

    auto outcome = s3Client.ListObjectsV2(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: listObjects: " <<
            outcome.GetError().GetMessage() << std::endl;
        return false;
    } else {
        Aws::Vector<Aws::S3::Model::Object> objects =
            outcome.GetResult().GetContents();

        allObjects.insert(allObjects.end(), objects.begin(), objects.end());
        continuationToken = outcome.GetResult().GetNextContinuationToken();
    }
} while (!continuationToken.empty());

std::cout << allObjects.size() << " object(s) found:" << std::endl;

for (const auto &object: allObjects) {
    std::cout << " " << object.GetKey() << std::endl;
    keysResult.push_back(object.GetKey());
}

return true;
}
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Um eine Liste von Objekten in einem Bucket zu erhalten

Das folgende `list-objects-v2` Beispiel listet die Objekte im angegebenen Bucket auf.

```
aws s3api list-objects-v2 \
```



```
--bucket my-bucket
```

Ausgabe:

```
{
  "Contents": [
    {
      "LastModified": "2019-11-05T23:11:50.000Z",
      "ETag": "\"621503c373607d548b37cff8778d992c\"",
      "StorageClass": "STANDARD",
      "Key": "doc1.rtf",
      "Size": 391
    },
    {
      "LastModified": "2019-11-05T23:11:50.000Z",
      "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",
      "StorageClass": "STANDARD",
      "Key": "doc2.rtf",
      "Size": 373
    },
    {
      "LastModified": "2019-11-05T23:11:50.000Z",
      "ETag": "\"08210852f65a2e9cb999972539a64d68\"",
      "StorageClass": "STANDARD",
      "Key": "doc3.rtf",
      "Size": 399
    },
    {
      "LastModified": "2019-11-05T23:11:50.000Z",
      "ETag": "\"d1852dd683f404306569471af106988e\"",
      "StorageClass": "STANDARD",
      "Key": "doc4.rtf",
      "Size": 6225
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS CLI Befehlsreferenz.

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// ListObjects lists the objects in a bucket.
func (basics BucketBasics) ListObjects(bucketName string) ([]types.Object, error)
{
    result, err := basics.S3Client.ListObjectsV2(context.TODO(),
        &s3.ListObjectsV2Input{
            Bucket: aws.String(bucketName),
        })
    var contents []types.Object
    if err != nil {
        log.Printf("Couldn't list objects in bucket %v. Here's why: %v\n", bucketName,
            err)
    } else {
        contents = result.Contents
    }
    return contents, err
}
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/**
 * Asynchronously lists all objects in the specified S3 bucket.
 *
 * @param bucketName the name of the S3 bucket to list objects for
 * @return a {@link CompletableFuture} that completes when all objects have
 * been listed
 */
public CompletableFuture<Void> listAllObjectsAsync(String bucketName) {
    ListObjectsV2Request initialRequest = ListObjectsV2Request.builder()
        .bucket(bucketName)
        .maxKeys(1)
        .build();

    ListObjectsV2Publisher paginator =
getAsyncClient().listObjectsV2Paginator(initialRequest);
    return paginator.subscribe(response -> {
        response.contents().forEach(s3Object -> {
            logger.info("Object key: " + s3Object.key());
        });
    }).thenRun(() -> {
        logger.info("Successfully listed all objects in the bucket: " +
bucketName);
    }).exceptionally(ex -> {
        throw new RuntimeException("Failed to list objects", ex);
    });
}
```

Listet Objekte mithilfe der Paginierung auf.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.paginators.ListObjectsV2Iterable;

public class ListObjectsPaginated {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName>\s

                Where:
                bucketName - The Amazon S3 bucket from which objects are
read.\s

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        listBucketObjects(s3, bucketName);
        s3.close();
    }

    public static void listBucketObjects(S3Client s3, String bucketName) {
        try {
            ListObjectsV2Request listReq = ListObjectsV2Request.builder()
                .bucket(bucketName)
                .maxKeys(1)
                .build();

            ListObjectsV2Iterable listRes = s3.listObjectsV2Paginator(listReq);
            listRes.stream()
                .flatMap(r -> r.contents().stream())

```

```
        .forEach(content -> System.out.println(" Key: " +
content.key() + " size = " + content.size())));

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Listen Sie alle Objekte in Ihrem Bucket auf. Wenn es mehr als ein Objekt gibt, `IsTruncated` `NextContinuationToken` wird es verwendet, um über die gesamte Liste zu iterieren.

```
import {
  S3Client,
  // This command supersedes the ListObjectsCommand and is the recommended way to
  list objects.
  ListObjectsV2Command,
} from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new ListObjectsV2Command({
    Bucket: "my-bucket",
    // The default and maximum number of keys returned is 1000. This limits it to
    // one for demonstration purposes.
    MaxKeys: 1,
  });
```

```
try {
  let isTruncated = true;

  console.log("Your bucket contains the following objects:\n");
  let contents = "";

  while (isTruncated) {
    const { Contents, IsTruncated, NextContinuationToken } =
      await client.send(command);
    const contentsList = Contents.map((c) => ` • ${c.Key}`).join("\n");
    contents += contentsList + "\n";
    isTruncated = IsTruncated;
    command.input.ContinuationToken = NextContinuationToken;
  }
  console.log(contents);
} catch (err) {
  console.error(err);
}
};
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun listBucketObjects(bucketName: String) {
  val request =
    ListObjectsRequest {
      bucket = bucketName
    }

  S3Client { region = "us-east-1" }.use { s3 ->
    val response = s3.listObjects(request)
  }
}
```

```
        response.contents?.forEach { myObject ->
            println("The name of the key is ${myObject.key}")
            println("The object is ${myObject.size?.let { calKb(it) }} KBs")
            println("The owner is ${myObject.owner}")
        }
    }
}

private fun calKb(intValue: Long): Long = intValue / 1024
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK als API Kotlin-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Listen Sie Objekte in einem Bucket auf.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

try {
    $contents = $this->s3client->listObjectsV2([
        'Bucket' => $this->bucketName,
    ]);
    echo "The contents of your bucket are: \n";
    foreach ($contents['Contents'] as $content) {
        echo $content['Key'] . "\n";
    }
} catch (Exception $exception) {
    echo "Failed to list objects in $this->bucketName with error: " .
    $exception->getMessage();
    exit("Please fix error with listing objects before continuing.");
}
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK for PHP API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
                               that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    @staticmethod
    def list(bucket, prefix=None):
        """
        Lists the objects in a bucket, optionally filtered by a prefix.

        :param bucket: The bucket to query. This is a Boto3 Bucket resource.
        :param prefix: When specified, only objects that start with this prefix
        are listed.
        :return: The list of objects.
        """
        try:
            if not prefix:
                objects = list(bucket.objects.all())
            else:
                objects = list(bucket.objects.filter(Prefix=prefix))
            logger.info(
```



```
        "Got objects %s from bucket '%s'", [o.key for o in objects],
        bucket.name
    )
    except ClientError:
        logger.exception("Couldn't get objects for bucket '%s'.",
        bucket.name)
        raise
    else:
        return objects
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK für Python (Boto3) API - Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketListObjectsWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
  def initialize(bucket)
    @bucket = bucket
  end

  # Lists object in a bucket.
  #
  # @param max_objects [Integer] The maximum number of objects to list.
  # @return [Integer] The number of objects listed.
  def list_objects(max_objects)
```

```

    count = 0
    puts "The objects in #{@bucket.name} are:"
    @bucket.objects.each do |obj|
      puts "\t#{obj.key}"
      count += 1
      break if count == max_objects
    end
    count
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list objects in bucket #{bucket.name}. Here's why:
#{e.message}"
    0
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"

  wrapper = BucketListObjectsWrapper.new(Aws::S3::Bucket.new(bucket_name))
  count = wrapper.list_objects(25)
  puts "Listed #{count} objects."
end

run_demo if $PROGRAM_NAME == __FILE__

```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

pub async fn list_objects(client: &aws_sdk_s3::Client, bucket: &str) ->
  Result<(), S3ExampleError> {
  let mut response = client

```

```

        .list_objects_v2()
        .bucket(bucket.to_owned())
        .max_keys(10) // In this example, go 10 at a time.
        .into_paginator()
        .send();

while let Some(result) = response.next().await {
    match result {
        Ok(output) => {
            for object in output.contents() {
                println!(" - {}", object.key().unwrap_or("Unknown"));
            }
        }
        Err(err) => {
            eprintln!("{err:?}")
        }
    }
}

Ok(())
}

```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK als API Referenz auf Rust.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

TRY.
    oo_result = lo_s3->listobjectsv2(           " oo_result is returned for
testing purposes. "
        iv_bucket = iv_bucket_name
    ).
    MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.

```

```
MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK als SAP ABAP API Referenz.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3

public func listBucketFiles(bucket: String) async throws -> [String] {
    do {
        let input = ListObjectsV2Input(
            bucket: bucket
        )

        // Use "Paginated" to get all the objects.
        // This lets the SDK handle the 'continuationToken' in
        "ListObjectsV2Output".
        let output = client.listObjectsV2Paginated(input: input)
        var names: [String] = []

        for try await page in output {
            guard let objList = page.contents else {
                print("ERROR: listObjectsV2Paginated returned nil contents.")
                continue
            }

            for obj in objList {
                if let objName = obj.key {
                    names.append(objName)
                }
            }
        }
    }
}
```

```
        return names
    }
    catch {
        print("ERROR: ", dump(error, name: "Listing objects. "))
        throw error
    }
}
```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in AWS SDK als API-Swift-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketAccelerateConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketAccelerateConfiguration`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// Amazon Simple Storage Service (Amazon S3) Transfer Acceleration is a
/// bucket-level feature that enables you to perform faster data transfers
/// to Amazon S3. This example shows how to configure Transfer
```

```
/// Acceleration.
/// </summary>
public class TransferAcceleration
{
    /// <summary>
    /// The main method initializes the client object and sets the
    /// Amazon Simple Storage Service (Amazon S3) bucket name before
    /// calling EnableAccelerationAsync.
    /// </summary>
    public static async Task Main()
    {
        var s3Client = new AmazonS3Client();
        const string bucketName = "amzn-s3-demo-bucket";

        await EnableAccelerationAsync(s3Client, bucketName);
    }

    /// <summary>
    /// This method sets the configuration to enable transfer acceleration
    /// for the bucket referred to in the bucketName parameter.
    /// </summary>
    /// <param name="client">An Amazon S3 client used to enable the
    /// acceleration on an Amazon S3 bucket.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket for which
the
    /// method will be enabling acceleration.</param>
    private static async Task EnableAccelerationAsync(AmazonS3Client client,
string bucketName)
    {
        try
        {
            var putRequest = new PutBucketAccelerateConfigurationRequest
            {
                BucketName = bucketName,
                AccelerateConfiguration = new AccelerateConfiguration
                {
                    Status = BucketAccelerateStatus.Enabled,
                },
            };
            await client.PutBucketAccelerateConfigurationAsync(putRequest);

            var getRequest = new GetBucketAccelerateConfigurationRequest
            {
                BucketName = bucketName,
```

```
        };
        var response = await
client.GetBucketAccelerateConfigurationAsync(getRequest);

        Console.WriteLine($"Acceleration state = '{response.Status}' ");
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error occurred. Message: '{ex.Message}' when
setting transfer acceleration");
    }
}
}
```

- API-Einheiten finden Sie [PutBucketAccelerateConfiguration](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um die Beschleunigungskonfiguration eines Buckets festzulegen

Das folgende `put-bucket-accelerate-configuration` Beispiel aktiviert die Accelerate-Konfiguration für den angegebenen Bucket.

```
aws s3api put-bucket-accelerate-configuration \
  --bucket my-bucket \
  --accelerate-configuration Status=Enabled
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketAccelerateConfiguration](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl aktiviert die Übertragungsbeschleunigung für den angegebenen S3-Bucket.

```
$statusVal = New-Object Amazon.S3.BucketAccelerateStatus('Enabled')  
Write-S3BucketAccelerateConfiguration -BucketName 's3testbucket' -  
AccelerateConfiguration_Status $statusVal
```

- API-Einheiten finden Sie [PutBucketAccelerateConfiguration](#) unter AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketAc1** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketAc1`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Zugriffskontrolllisten verwalten \(\) ACLs](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).


```
/// <summary>
/// Creates an Amazon S3 bucket with an ACL to control access to the
/// bucket and the objects stored in it.
/// </summary>
/// <param name="client">The initialized client object used to create
/// an Amazon S3 bucket, with an ACL applied to the bucket.
/// </param>
/// <param name="region">The AWS Region where the bucket will be
created.</param>
/// <param name="newBucketName">The name of the bucket to create.</param>
/// <returns>A boolean value indicating success or failure.</returns>
public static async Task<bool> CreateBucketUseCannedACLAsync(IAmazonS3
client, S3Region region, string newBucketName)
{
    try
    {
        // Create a new Amazon S3 bucket with Canned ACL.
        var putBucketRequest = new PutBucketRequest()
        {
            BucketName = newBucketName,
            BucketRegion = region,
            CannedACL = S3CannedACL.LogDeliveryWrite,
        };

        PutBucketResponse putBucketResponse = await
client.PutBucketAsync(putBucketRequest);

        return putBucketResponse.HttpStatusCode ==
System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Amazon S3 error: {ex.Message}");
    }

    return false;
}
```

- API-Einheiten finden Sie [PutBucketAcl](#) unter AWS SDK for .NET API-Referenz.

C++

SDK für C++

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::putBucketAcl(const Aws::String &bucketName, const Aws::String
&ownerID,
                                const Aws::String &granteePermission,
                                const Aws::String &granteeType, const Aws::String
&granteeID,
                                const Aws::String &granteeEmailAddress,
                                const Aws::String &granteeURI, const
Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::Owner owner;
    owner.SetID(ownerID);

    Aws::S3::Model::Grantee grantee;
    grantee.SetType(setGranteeType(granteeType));

    if (!granteeEmailAddress.empty()) {
        grantee.SetEmailAddress(granteeEmailAddress);
    }

    if (!granteeID.empty()) {
        grantee.SetID(granteeID);
    }

    if (!granteeURI.empty()) {
        grantee.SetURI(granteeURI);
    }

    Aws::S3::Model::Grant grant;
    grant.SetGrantee(grantee);
    grant.SetPermission(setGranteePermission(granteePermission));
```

```
Aws::Vector<Aws::S3::Model::Grant> grants;
grants.push_back(grant);

Aws::S3::Model::AccessControlPolicy acp;
acp.SetOwner(owner);
acp.SetGrants(grants);

Aws::S3::Model::PutBucketAclRequest request;
request.SetAccessControlPolicy(acp);
request.SetBucket(bucketName);

Aws::S3::Model::PutBucketAclOutcome outcome =
    s3Client.PutBucketAcl(request);

if (!outcome.IsSuccess()) {
    const Aws::S3::S3Error &error = outcome.GetError();

    std::cerr << "Error: putBucketAcl: " << error.GetExceptionName()
                << " - " << error.GetMessage() << std::endl;
} else {
    std::cout << "Successfully added an ACL to the bucket '" << bucketName
                << "'." << std::endl;
}

return outcome.IsSuccess();
}

//! Routine which converts a human-readable string to a built-in type
enumeration.
/*!
 \param access: Human readable string.
 \return Permission: A Permission enum.
 */

Aws::S3::Model::Permission setGranteePermission(const Aws::String &access) {
    if (access == "FULL_CONTROL")
        return Aws::S3::Model::Permission::FULL_CONTROL;
    if (access == "WRITE")
        return Aws::S3::Model::Permission::WRITE;
    if (access == "READ")
        return Aws::S3::Model::Permission::READ;
    if (access == "WRITE_ACP")
        return Aws::S3::Model::Permission::WRITE_ACP;
    if (access == "READ_ACP")
```

```

        return Aws::S3::Model::Permission::READ_ACP;
        return Aws::S3::Model::Permission::NOT_SET;
    }

    //! Routine which converts a human-readable string to a built-in type
    enumeration.
    /*!
    \param type: Human readable string.
    \return Type: Type enumeration
    */

    Aws::S3::Model::Type setGranteeType(const Aws::String &type) {
        if (type == "Amazon customer by email")
            return Aws::S3::Model::Type::AmazonCustomerByEmail;
        if (type == "Canonical user")
            return Aws::S3::Model::Type::CanonicalUser;
        if (type == "Group")
            return Aws::S3::Model::Type::Group;
        return Aws::S3::Model::Type::NOT_SET;
    }

```

- API-Einheiten finden Sie [PutBucketAcl](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Dieses Beispiel gewährt `full control` zwei AWS Benutzer (`user1@example.com` und `user2@example.com`) und `read` Erlaubnis für alle:

```
aws s3api put-bucket-acl --bucket MyBucket --grant-full-
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Einheiten zu Benutzerdefiniert finden Sie unter <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> ACLs (die ACL `s3api`-Befehle verwenden z. `put-bucket-acl` B. dieselbe Kurzschreibweise für Argumente).

- Einzelheiten finden Sie unter API [PutBucketAcl](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AccessControlPolicy;
import software.amazon.awssdk.services.s3.model.Grant;
import software.amazon.awssdk.services.s3.model.Permission;
import software.amazon.awssdk.services.s3.model.PutBucketAclRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.Type;

import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class SetAcl {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName> <id>\s

                Where:
                bucketName - The Amazon S3 bucket to grant permissions on.\s
                id - The ID of the owner of this bucket (you can get this value
                from the AWS Management Console).
```

```
        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String bucketName = args[0];
    String id = args[1];
    System.out.format("Setting access \n");
    System.out.println(" in bucket: " + bucketName);
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    setBucketAcl(s3, bucketName, id);
    System.out.println("Done!");
    s3.close();
}

public static void setBucketAcl(S3Client s3, String bucketName, String id) {
    try {
        Grant ownerGrant = Grant.builder()
            .grantee(builder -> builder.id(id)
                .type(Type.CANONICAL_USER))
            .permission(Permission.FULL_CONTROL)
            .build();

        List<Grant> grantList2 = new ArrayList<>();
        grantList2.add(ownerGrant);

        AccessControlPolicy acl = AccessControlPolicy.builder()
            .owner(builder -> builder.id(id))
            .grants(grantList2)
            .build();

        PutBucketAclRequest putAclReq = PutBucketAclRequest.builder()
            .bucket(bucketName)
            .accessControlPolicy(acl)
            .build();

        s3.putBucketAcl(putAclReq);
    }
}
```

```
    } catch (S3Exception e) {  
        e.printStackTrace();  
        System.exit(1);  
    }  
}  
}
```

- API Einzelheiten finden Sie [PutBucketAcl](#) unter AWS SDK for Java 2.x API Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Stell den Eimer hinACL.

```
import { PutBucketAclCommand, S3Client } from "@aws-sdk/client-s3";  
  
const client = new S3Client({});  
  
// Most Amazon S3 use cases don't require the use of access control lists (ACLs).  
// We recommend that you disable ACLs, except in unusual circumstances where  
// you need to control access for each object individually.  
// Consider a policy instead. For more information see https://  
docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html.  
export const main = async () => {  
    // Grant a user READ access to a bucket.  
    const command = new PutBucketAclCommand({  
        Bucket: "test-bucket",  
        AccessControlPolicy: {  
            Grants: [  
                {  
                    Grantee: {  
                        // The canonical ID of the user. This ID is an obfuscated form of  
                        your AWS account number.  
                        // It's unique to Amazon S3 and can't be found elsewhere.  
                    }  
                }  
            ]  
        }  
    });  
};
```

```
        // For more information, see https://docs.aws.amazon.com/AmazonS3/
latest/userguide/finding-canonical-user-id.html.
        ID: "canonical-id-1",
        Type: "CanonicalUser",
    },
    // One of FULL_CONTROL | READ | WRITE | READ_ACP | WRITE_ACP
    // https://docs.aws.amazon.com/AmazonS3/latest/API/
API_Grant.html#AmazonS3-Type-Grant-Permission
    Permission: "FULL_CONTROL",
},
],
Owner: {
    ID: "canonical-id-2",
},
},
});

try {
    const response = await client.send(command);
    console.log(response);
} catch (err) {
    console.error(err);
}
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [PutBucketAcl](#) in AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun setBucketAcl(
    bucketName: String,
```



```
        idVal: String,
    ) {
        val myGrant =
            Grantee {
                id = idVal
                type = Type.CanonicalUser
            }

        val ownerGrant =
            Grant {
                grantee = myGrant
                permission = Permission.FullControl
            }

        val grantList = mutableListOf<Grant>()
        grantList.add(ownerGrant)

        val ownerOb =
            Owner {
                id = idVal
            }

        val acl =
            AccessControlPolicy {
                owner = ownerOb
                grants = grantList
            }

        val request =
            PutBucketAclRequest {
                bucket = bucketName
                accessControlPolicy = acl
            }

        S3Client { region = "us-east-1" }.use { s3 ->
            s3.putBucketAcl(request)
            println("An ACL was successfully set on $bucketName")
        }
    }
}
```

- API-Einheiten finden Sie [PutBucketAcl](#) unter AWS SDK als API Kotlin-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
            that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def grant_log_delivery_access(self):
        """
        Grant the AWS Log Delivery group write access to the bucket so that
        Amazon S3 can deliver access logs to the bucket. This is the only
        recommended
        use of an S3 bucket ACL.
        """
        try:
            acl = self.bucket.Acl()
            # Putting an ACL overwrites the existing ACL. If you want to preserve
            # existing grants, append new grants to the list of existing grants.
            grants = acl.grants if acl.grants else []
            grants.append(
                {
                    "Grantee": {
                        "Type": "Group",
                        "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery",
                    },
                    "Permission": "WRITE",
```

```
        }
    )
    acl.put(AccessControlPolicy={"Grants": grants, "Owner": acl.owner})
    logger.info("Granted log delivery access to bucket '%s'",
self.bucket.name)
    except ClientError:
        logger.exception("Couldn't add ACL to bucket '%s'.",
self.bucket.name)
        raise
```

- API-Einheiten finden Sie [PutBucketAcl](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketCors** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketCors`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Add CORS configuration to the Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used
/// to apply the CORS configuration to an Amazon S3 bucket.</param>
/// <param name="configuration">The CORS configuration to apply.</param>
private static async Task PutCORSConfigurationAsync(AmazonS3Client
client, CORSConfiguration configuration)
```

```
{
    PutCORSConfigurationRequest request = new
PutCORSConfigurationRequest()
    {
        BucketName = BucketName,
        Configuration = configuration,
    };

    _ = await client.PutCORSConfigurationAsync(request);
}
```

- API Einzelheiten finden Sie [PutBucketCors](#) unter AWS SDK for .NET API Referenz.

CLI

AWS CLI

Das folgende Beispiel aktiviert PUT/POST, und DELETE Anfragen von `www.example.com` und ermöglicht GET Anfragen von einer beliebigen Domain:

```
aws s3api put-bucket-cors --bucket MyBucket --cors-configuration file://cors.json
```

cors.json:

```
{
  "CORSRules": [
    {
      "AllowedOrigins": ["http://www.example.com"],
      "AllowedHeaders": ["*"],
      "AllowedMethods": ["PUT", "POST", "DELETE"],
      "MaxAgeSeconds": 3000,
      "ExposeHeaders": ["x-amz-server-side-encryption"]
    },
    {
      "AllowedOrigins": ["*"],
      "AllowedHeaders": ["Authorization"],
      "AllowedMethods": ["GET"],
      "MaxAgeSeconds": 3000
    }
  ]
}
```

- API-Einheiten finden Sie unter [PutBucketCors](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import java.util.ArrayList;
import java.util.List;
import software.amazon.awssdk.services.s3.model.GetBucketCorsRequest;
import software.amazon.awssdk.services.s3.model.GetBucketCorsResponse;
import software.amazon.awssdk.services.s3.model.DeleteBucketCorsRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.CORSRule;
import software.amazon.awssdk.services.s3.model.CORSConfiguration;
import software.amazon.awssdk.services.s3.model.PutBucketCorsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class S3Cors {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <accountId>\s

            Where:
                bucketName - The Amazon S3 bucket to upload an object into.
```

```
        accountId - The id of the account that owns the Amazon S3
bucket.

        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String bucketName = args[0];
    String accountId = args[1];
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    setCorsInformation(s3, bucketName, accountId);
    getBucketCorsInformation(s3, bucketName, accountId);
    deleteBucketCorsInformation(s3, bucketName, accountId);
    s3.close();
}

public static void deleteBucketCorsInformation(S3Client s3, String
bucketName, String accountId) {
    try {
        DeleteBucketCorsRequest bucketCorsRequest =
DeleteBucketCorsRequest.builder()
        .bucket(bucketName)
        .expectedBucketOwner(accountId)
        .build();

        s3.deleteBucketCors(bucketCorsRequest);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void getBucketCorsInformation(S3Client s3, String bucketName,
String accountId) {
    try {
        GetBucketCorsRequest bucketCorsRequest =
GetBucketCorsRequest.builder()
```

```
        .bucket(bucketName)
        .expectedBucketOwner(accountId)
        .build();

    GetBucketCorsResponse corsResponse =
s3.getBucketCors(bucketCorsRequest);
    List<CORSRule> corsRules = corsResponse.corsRules();
    for (CORSRule rule : corsRules) {
        System.out.println("allowOrigins: " + rule.allowedOrigins());
        System.out.println("AllowedMethod: " + rule.allowedMethods());
    }

} catch (S3Exception e) {

    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}

}

public static void setCorsInformation(S3Client s3, String bucketName, String
accountId) {
    List<String> allowMethods = new ArrayList<>();
    allowMethods.add("PUT");
    allowMethods.add("POST");
    allowMethods.add("DELETE");

    List<String> allowOrigins = new ArrayList<>();
    allowOrigins.add("http://example.com");
    try {
        // Define CORS rules.
        CORSRule corsRule = CORSRule.builder()
            .allowedMethods(allowMethods)
            .allowedOrigins(allowOrigins)
            .build();

        List<CORSRule> corsRules = new ArrayList<>();
        corsRules.add(corsRule);
        CORSConfiguration configuration = CORSConfiguration.builder()
            .corsRules(corsRules)
            .build();

        PutBucketCorsRequest putBucketCorsRequest =
PutBucketCorsRequest.builder()
            .bucket(bucketName)
```

```
        .corsConfiguration(configuration)
        .expectedBucketOwner(accountId)
        .build();

    s3.putBucketCors(putBucketCorsRequest);

} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

- API-Einheiten finden Sie [PutBucketCors](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Fügen Sie eine CORS-Regel hinzu.

```
import { PutBucketCorsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// By default, Amazon S3 doesn't allow cross-origin requests. Use this command
// to explicitly allow cross-origin requests.
export const main = async () => {
    const command = new PutBucketCorsCommand({
        Bucket: "test-bucket",
        CORSConfiguration: {
            CORSRules: [
                {
                    // Allow all headers to be sent to this bucket.
                    AllowedHeaders: ["*"],
                }
            ]
        }
    });
};
```



```

    // Allow only GET and PUT methods to be sent to this bucket.
    AllowedMethods: ["GET", "PUT"],
    // Allow only requests from the specified origin.
    AllowedOrigins: ["https://www.example.com"],
    // Allow the entity tag (ETag) header to be returned in the response.
The ETag header
    // The entity tag represents a specific version of the object. The ETag
reflects
    // changes only to the contents of an object, not its metadata.
    ExposeHeaders: ["ETag"],
    // How long the requesting browser should cache the preflight response.
After
    // this time, the preflight request will have to be made again.
    MaxAgeSeconds: 3600,
  },
],
},
});

try {
  const response = await client.send(command);
  console.log(response);
} catch (err) {
  console.error(err);
}
};

```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [PutBucketCors](#) in AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
                        that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def put_cors(self, cors_rules):
        """
        Apply CORS rules to the bucket. CORS rules specify the HTTP actions that
        are
        allowed from other domains.

        :param cors_rules: The CORS rules to apply.
        """
        try:
            self.bucket.Cors().put(CORSConfiguration={"CORSRules": cors_rules})
            logger.info(
                "Put CORS rules %s for bucket '%s'.", cors_rules,
                self.bucket.name
            )
        except ClientError:
            logger.exception("Couldn't put CORS rules for bucket %s.",
                self.bucket.name)
            raise
```

- API Einzelheiten finden Sie [PutBucketCors](#) unter AWS SDK für Python (Boto3) API -Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
  # an existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Sets CORS rules on a bucket.
  #
  # @param allowed_methods [Array<String>] The types of HTTP requests to allow.
  # @param allowed_origins [Array<String>] The origins to allow.
  # @returns [Boolean] True if the CORS rules were set; otherwise, false.
  def set_cors(allowed_methods, allowed_origins)
    @bucket_cors.put(
      cors_configuration: {
        cors_rules: [
          {
            allowed_methods: allowed_methods,
            allowed_origins: allowed_origins,
            allowed_headers: %w[*],
            max_age_seconds: 3600
          }
        ]
      }
    )
    true
  rescue Aws::Errors::ServiceError => e
```

```

    puts "Couldn't set CORS rules for #{@bucket_cors.bucket.name}. Here's why:
    #{e.message}"
    false
  end
end
end

```

- API Einzelheiten finden Sie [PutBucketCors](#) unter AWS SDK for Ruby API Referenz.

Für eine vollständige Liste von AWS SDK Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **PutBucketEncryption** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt PutBucketEncryption.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

    /// <summary>
    /// Set the bucket server side encryption to use AWSKMS with a customer-
    managed key id.
    /// </summary>
    /// <param name="bucketName">Name of the bucket.</param>
    /// <param name="kmsKeyId">The Id of the KMS Key.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> SetBucketServerSideEncryption(string
    bucketName, string kmsKeyId)
    {
        var serverSideEncryptionByDefault = new ServerSideEncryptionConfiguration
        {
            ServerSideEncryptionRules = new List<ServerSideEncryptionRule>

```

```
        {
            new ServerSideEncryptionRule
            {
                ServerSideEncryptionByDefault = new
ServerSideEncryptionByDefault
                {
                    ServerSideEncryptionAlgorithm =
ServerSideEncryptionMethod.AWSKMS,
                    ServerSideEncryptionKeyManagementServiceKeyId = kmsKeyId
                }
            }
        };
    try
    {
        var encryptionResponse = await _s3Client.PutBucketEncryptionAsync(new
PutBucketEncryptionRequest
        {
            BucketName = bucketName,
            ServerSideEncryptionConfiguration =
serverSideEncryptionByDefault,
        });

        return encryptionResponse.HttpStatusCode == HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine(ex.ErrorCode == "AccessDenied"
            ? $"This account does not have permission to set encryption on
{bucketName}, please try again."
            : $"Unable to set bucket encryption for bucket {bucketName},
{ex.Message}");
    }
    return false;
}
```

- API-Einheiten finden Sie [PutBucketEncryption](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um die serverseitige Verschlüsselung für einen Bucket zu konfigurieren

Im folgenden `put-bucket-encryption` Beispiel wird die AES256 Verschlüsselung als Standard für den angegebenen Bucket festgelegt.

```
aws s3api put-bucket-encryption \  
  --bucket my-bucket \  
  --server-side-encryption-configuration '{"Rules":  
  [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketEncryption](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl aktiviert die AES256 serverseitige Standardverschlüsselung mit Amazon S3 Managed Keys (SSE-S3) für den angegebenen Bucket.

```
$Encryptionconfig = @{ServerSideEncryptionByDefault =  
  @{ServerSideEncryptionAlgorithm = "AES256"}}  
Set-S3BucketEncryption -BucketName 's3testbucket' -  
  ServerSideEncryptionConfiguration_ServerSideEncryptionRule $Encryptionconfig
```

- API-Einheiten finden Sie unter [PutBucketEncryption](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketLifecycleConfiguration** mit einem AWS SDK oder CLI


Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketLifecycleConfiguration`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Löschen Sie unvollständige mehrteilige Uploads](#)
- [Mit versionierten Objekten arbeiten](#)

.NET

AWS SDK for .NET

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Adds lifecycle configuration information to the S3 bucket named in
/// the bucketName parameter.
/// </summary>
/// <param name="client">The S3 client used to call the
/// PutLifecycleConfigurationAsync method.</param>
/// <param name="bucketName">A string representing the S3 bucket to
/// which configuration information will be added.</param>
/// <param name="configuration">A LifecycleConfiguration object that
/// will be applied to the S3 bucket.</param>
public static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
string bucketName, LifecycleConfiguration configuration)
{
    var request = new PutLifecycleConfigurationRequest()
    {
        BucketName = bucketName,
        Configuration = configuration,
    };
    var response = await client.PutLifecycleConfigurationAsync(request);
}
```

- API-Einheiten finden Sie [PutBucketLifecycleConfiguration](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Der folgende Befehl wendet eine Lebenszykluskonfiguration auf einen Bucket mit dem Namen `army-bucket`:

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-configuration file://lifecycle.json
```

Die Datei `lifecycle.json` ist ein JSON-Dokument im aktuellen Ordner, das zwei Regeln festlegt:

```
{
  "Rules": [
    {
      "ID": "Move rotated logs to Glacier",
      "Prefix": "rotated/",
      "Status": "Enabled",
      "Transitions": [
        {
          "Date": "2015-11-10T00:00:00.000Z",
          "StorageClass": "GLACIER"
        }
      ]
    },
    {
      "Status": "Enabled",
      "Prefix": "",
      "NoncurrentVersionTransitions": [
        {
          "NoncurrentDays": 2,
          "StorageClass": "GLACIER"
        }
      ],
      "ID": "Move old versions to Glacier"
    }
  ]
}
```



```
}
```

Die erste Regel verschiebt Dateien mit dem Präfix am angegebenen Datum `rotated` nach Glacier. Die zweite Regel verschiebt alte Objektversionen nach Glacier, wenn sie nicht mehr aktuell sind. Informationen zu akzeptablen Zeitstempelformaten finden Sie unter Angeben von Parameterwerten im AWS CLI-Benutzerhandbuch.

- API-Einheiten finden Sie in [PutBucketLifecycleConfiguration](#) AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.LifecycleRuleFilter;
import software.amazon.awssdk.services.s3.model.Transition;
import
    software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationRequest;
import
    software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationResponse;
import software.amazon.awssdk.services.s3.model.DeleteBucketLifecycleRequest;
import software.amazon.awssdk.services.s3.model.TransitionStorageClass;
import software.amazon.awssdk.services.s3.model.LifecycleRule;
import software.amazon.awssdk.services.s3.model.ExpirationStatus;
import software.amazon.awssdk.services.s3.model.BucketLifecycleConfiguration;
import
    software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/

public class LifecycleConfiguration {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <accountId>\s

            Where:
                bucketName - The Amazon Simple Storage Service
(Amazon S3) bucket to upload an object into.
                accountId - The id of the account that owns the
Amazon S3 bucket.

            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String accountId = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        setLifecycleConfig(s3, bucketName, accountId);
        getLifecycleConfig(s3, bucketName, accountId);
        deleteLifecycleConfig(s3, bucketName, accountId);
        System.out.println("You have successfully created, updated, and
deleted a Lifecycle configuration");
        s3.close();
    }

    public static void setLifecycleConfig(S3Client s3, String bucketName,
String accountId) {
        try {
```

```
        // Create a rule to archive objects with the
"glacierobjects/" prefix to Amazon
        // S3 Glacier.
        LifecycleRuleFilter ruleFilter =
LifecycleRuleFilter.builder()
                                .prefix("glacierobjects/")
                                .build();

        Transition transition = Transition.builder()

.storageClass(TransitionStorageClass.GLACIER)
                .days(0)
                .build();

        LifecycleRule rule1 = LifecycleRule.builder()
                .id("Archive immediately rule")
                .filter(ruleFilter)
                .transitions(transition)
                .status(ExpirationStatus.ENABLED)
                .build();

        // Create a second rule.
        Transition transition2 = Transition.builder()

.storageClass(TransitionStorageClass.GLACIER)
                .days(0)
                .build();

        List<Transition> transitionList = new ArrayList<>();
        transitionList.add(transition2);

        LifecycleRuleFilter ruleFilter2 =
LifecycleRuleFilter.builder()
                                .prefix("glacierobjects/")
                                .build();

        LifecycleRule rule2 = LifecycleRule.builder()
                .id("Archive and then delete rule")
                .filter(ruleFilter2)
                .transitions(transitionList)
                .status(ExpirationStatus.ENABLED)
                .build();

        // Add the LifecycleRule objects to an ArrayList.
```

```
        ArrayList<LifecycleRule> ruleList = new ArrayList<>();
        ruleList.add(rule1);
        ruleList.add(rule2);

        BucketLifecycleConfiguration lifecycleConfiguration =
BucketLifecycleConfiguration.builder()
                                .rules(ruleList)
                                .build();

        PutBucketLifecycleConfigurationRequest
putBucketLifecycleConfigurationRequest = PutBucketLifecycleConfigurationRequest
                                            .builder()
                                            .bucket(bucketName)

.lifecycleConfiguration(lifecycleConfiguration)
                                .expectedBucketOwner(accountId)
                                .build();

s3.putBucketLifecycleConfiguration(putBucketLifecycleConfigurationRequest);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Retrieve the configuration and add a new rule.
    public static void getLifecycleConfig(S3Client s3, String bucketName,
String accountId) {
        try {
            GetBucketLifecycleConfigurationRequest
getBucketLifecycleConfigurationRequest = GetBucketLifecycleConfigurationRequest
                                            .builder()
                                            .bucket(bucketName)
                                            .expectedBucketOwner(accountId)
                                            .build();

            GetBucketLifecycleConfigurationResponse response = s3

.lifecycleConfiguration(getBucketLifecycleConfigurationRequest);
            List<LifecycleRule> newList = new ArrayList<>();
            List<LifecycleRule> rules = response.rules();
            for (LifecycleRule rule : rules) {
```

```
        newList.add(rule);
    }

    // Add a new rule with both a prefix predicate and a tag
    predicate.
        LifecycleRuleFilter ruleFilter =
LifecycleRuleFilter.builder()
            .prefix("YearlyDocuments/")
            .build();

        Transition transition = Transition.builder()

.storageClass(TransitionStorageClass.GLACIER)
            .days(3650)
            .build();

        LifecycleRule rule1 = LifecycleRule.builder()
            .id("NewRule")
            .filter(ruleFilter)
            .transitions(transition)
            .status(ExpirationStatus.ENABLED)
            .build();

        // Add the new rule to the list.
        newList.add(rule1);
        BucketLifecycleConfiguration lifecycleConfiguration =
BucketLifecycleConfiguration.builder()
            .rules(newList)
            .build();

        PutBucketLifecycleConfigurationRequest
putBucketLifecycleConfigurationRequest = PutBucketLifecycleConfigurationRequest
            .builder()
            .bucket(bucketName)

.lifecycleConfiguration(lifecycleConfiguration)
            .expectedBucketOwner(accountId)
            .build();

s3.putBucketLifecycleConfiguration(putBucketLifecycleConfigurationRequest);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
    }
```

```
        System.exit(1);
    }
}

// Delete the configuration from the Amazon S3 bucket.
public static void deleteLifecycleConfig(S3Client s3, String bucketName,
String accountId) {
    try {
        DeleteBucketLifecycleRequest deleteBucketLifecycleRequest
= DeleteBucketLifecycleRequest
                                .builder()
                                .bucket(bucketName)
                                .expectedBucketOwner(accountId)
                                .build();

        s3.deleteBucketLifecycle(deleteBucketLifecycleRequest);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- API-Einheiten finden Sie [PutBucketLifecycleConfiguration](#) unter AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""
```

```
def __init__(self, bucket):
    """
    :param bucket: A Boto3 Bucket resource. This is a high-level resource in
    Boto3
                   that wraps bucket actions in a class-like structure.
    """
    self.bucket = bucket
    self.name = bucket.name

def put_lifecycle_configuration(self, lifecycle_rules):
    """
    Apply a lifecycle configuration to the bucket. The lifecycle
    configuration can
    be used to archive or delete the objects in the bucket according to
    specified
    parameters, such as a number of days.

    :param lifecycle_rules: The lifecycle rules to apply.
    """
    try:
        self.bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={"Rules": lifecycle_rules}
        )
        logger.info(
            "Put lifecycle rules %s for bucket '%s'.",
            lifecycle_rules,
            self.bucket.name,
        )
    except ClientError:
        logger.exception(
            "Couldn't put lifecycle rules for bucket '%s'.", self.bucket.name
        )
        raise
```

- API-Einheiten finden Sie [PutBucketLifecycleConfiguration](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketLogging** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketLogging`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
/// this example. The first is the bucket for which you wish to enable
/// logging, and the second is the location where you want to store the
/// logs.
/// </summary>
public class ServerAccessLogging
{
    private static IConfiguration _configuration = null!;

    public static async Task Main()
    {
        LoadConfig();

        string bucketName = _configuration["BucketName"];
        string logBucketName = _configuration["LogBucketName"];
    }
}
```



```
string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
string accountId = _configuration["AccountId"];

// If the AWS Region defined for your default user is different
// from the Region where your Amazon S3 bucket is located,
// pass the Region name to the Amazon S3 client object's constructor.
// For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
IAmazonS3 client = new AmazonS3Client();

try
{
    // Update bucket policy for target bucket to allow delivery of
logs to it.
    await SetBucketPolicyToAllowLogDelivery(
        client,
        bucketName,
        logBucketName,
        logObjectKeyPrefix,
        accountId);

    // Enable logging on the source bucket.
    await EnableLoggingAsync(
        client,
        bucketName,
        logBucketName,
        logObjectKeyPrefix);
}
catch (AmazonS3Exception e)
{
    Console.WriteLine($"Error: {e.Message}");
}
}

/// <summary>
/// This method grants appropriate permissions for logging to the
/// Amazon S3 bucket where the logs will be stored.
/// </summary>
/// <param name="client">The initialized Amazon S3 client which will be
used
/// to apply the bucket policy.</param>
/// <param name="sourceBucketName">The name of the source bucket.</param>
/// <param name="logBucketName">The name of the bucket where logging
/// information will be stored.</param>
```

```

    /// <param name="logPrefix">The logging prefix where the logs should be
    delivered.</param>
    /// <param name="accountId">The account id of the account where the
    source bucket exists.</param>
    /// <returns>Async task.</returns>
    public static async Task SetBucketPolicyToAllowLogDelivery(
        IAmazonS3 client,
        string sourceBucketName,
        string logBucketName,
        string logPrefix,
        string accountId)
    {
        var resourceArn = @""arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @"""";

        var newPolicy = @"{
                                ""Statement"": [{
                                ""Sid"": ""S3ServerAccessLogsPolicy"",
                                ""Effect"": ""Allow"",
                                ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
                                ""Action"": [""s3:PutObject""],
                                ""Resource"": ["" + resourceArn + @""],
                                ""Condition"": {
                                ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @"""" },
                                ""StringEquals"": { ""aws:SourceAccount"": """" +
accountId + @"""" }
                                }
                                }
                                }];

        Console.WriteLine($"The policy to apply to bucket {logBucketName} to
enable logging:");
        Console.WriteLine(newPolicy);

        PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
        {
            BucketName = logBucketName,
            Policy = newPolicy,
        };
        await client.PutBucketPolicyAsync(putRequest);
        Console.WriteLine("Policy applied.");
    }

```

```
    /// <summary>
    /// This method enables logging for an Amazon S3 bucket. Logs will be
stored
    /// in the bucket you selected for logging. Selected prefix
    /// will be prepended to each log object.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client which will be
used
    /// to configure and apply logging to the selected Amazon S3 bucket.</
param>
    /// <param name="bucketName">The name of the Amazon S3 bucket for which
you
    /// wish to enable logging.</param>
    /// <param name="logBucketName">The name of the Amazon S3 bucket where
logging
    /// information will be stored.</param>
    /// <param name="logObjectKeyPrefix">The prefix to prepend to each
    /// object key.</param>
    /// <returns>Async task.</returns>
    public static async Task EnableLoggingAsync(
        IAmazonS3 client,
        string bucketName,
        string logBucketName,
        string logObjectKeyPrefix)
    {
        Console.WriteLine($"Enabling logging for bucket {bucketName}.");
        var loggingConfig = new S3BucketLoggingConfig
        {
            TargetBucketName = logBucketName,
            TargetPrefix = logObjectKeyPrefix,
        };

        var putBucketLoggingRequest = new PutBucketLoggingRequest
        {
            BucketName = bucketName,
            LoggingConfig = loggingConfig,
        };
        await client.PutBucketLoggingAsync(putBucketLoggingRequest);
        Console.WriteLine($"Logging enabled.");
    }

    /// <summary>
    /// Loads configuration from settings files.
    /// </summary>
```

```
public static void LoadConfig()
{
    _configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load settings from .json file.
        .AddJsonFile("settings.local.json", true) // Optionally, load
local settings.
        .Build();
}
}
```

- API-Einheiten finden Sie [PutBucketLogging](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Beispiel 1: So legen Sie die Bucket-Policy-Protokollierung fest

Im folgenden `put-bucket-logging` Beispiel wird die Protokollierungsrichtlinie für festgelegt `MyBucket`. Erteilen Sie zunächst mit dem `put-bucket-policy` Befehl dem Prinzipal des Logging-Dienstes die Berechtigung in Ihrer Bucket-Richtlinie.

```
aws s3api put-bucket-policy \
  --bucket MyBucket \
  --policy file://policy.json
```

Inhalt von `policy.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3ServerAccessLogsPolicy",
      "Effect": "Allow",
      "Principal": {"Service": "logging.s3.amazonaws.com"},
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::MyBucket/Logs/*",
      "Condition": {
```

```

        "ArnLike": {"aws:SourceARN": "arn:aws:s3:::SOURCE-BUCKET-NAME"},
        "StringEquals": {"aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"}
    }
}
]
}

```

Um die Protokollierungsrichtlinie anzuwenden, verwenden Sie `put-bucket-logging`.

```

aws s3api put-bucket-logging \
  --bucket MyBucket \
  --bucket-logging-status file:///logging.json

```

Inhalt von `logging.json`:

```

{
  "LoggingEnabled": {
    "TargetBucket": "MyBucket",
    "TargetPrefix": "Logs/"
  }
}

```

Der `put-bucket-policy` Befehl ist erforderlich, um dem Prinzipal des Protokollierungsdienstes `s3:PutObject` Berechtigungen zu erteilen.

Weitere Informationen finden Sie unter [Amazon S3 Server Access Logging](#) im Amazon S3 S3-Benutzerhandbuch.

Beispiel 2: So legen Sie eine Bucket-Richtlinie für die Protokollierung des Zugriffs auf nur einen einzelnen Benutzer fest

Im folgenden `put-bucket-logging` Beispiel wird die Protokollierungsrichtlinie für festgelegt `MyBucket`. Das Tool AWS Der Benutzer `bob@example.com` hat die volle Kontrolle über die Protokolldateien, und niemand sonst hat Zugriff darauf. Erteilen Sie zunächst die S3-Erlaubnis mit `put-bucket-acl`.

```

aws s3api put-bucket-acl \
  --bucket MyBucket \
  --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery \
  --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery

```

Wenden Sie dann die Protokollierungsrichtlinie mit `aws s3api put-bucket-logging`.

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --bucket-logging-status file:///logging.json
```

Inhalt von `logging.json`:

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "MyBucket",  
    "TargetPrefix": "MyBucketLogs/",  
    "TargetGrants": [  
      {  
        "Grantee": {  
          "Type": "AmazonCustomerByEmail",  
          "EmailAddress": "bob@example.com"  
        },  
        "Permission": "FULL_CONTROL"  
      }  
    ]  
  }  
}
```

Der `put-bucket-acl` Befehl ist erforderlich, um dem Protokollübermittlungssystem von S3 die erforderlichen Berechtigungen (Schreib- und Lese-ACP-Berechtigungen) zu gewähren.

Weitere Informationen finden Sie unter [Amazon S3 Server Access Logging](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie [PutBucketLogging](#) unter AWS CLI Befehlsreferenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketNotification** mit einem AWS SDK oder CLI.

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketNotification`.

CLI

AWS CLI

Das wendet eine Benachrichtigungskonfiguration auf einen Bucket mit dem Namen `army-bucket`:

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration file://notification.json
```

Die Datei `notification.json` ist ein JSON Dokument im aktuellen Ordner, das ein SNS Thema und einen Ereignistyp angibt, die überwacht werden sollen:

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

Dem SNS Thema muss eine IAM Richtlinie beigefügt sein, die es Amazon S3 ermöglicht, darin zu veröffentlichen:

```
{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:us-west-2:123456789012:my-bucket",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
        }
      }
    }
  ]
}
```

```
}  
]  
}
```

- API-Einheiten finden Sie [PutBucketNotification](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird die SNS Themenkonfiguration für das S3-Ereignis `ObjectRemovedDelete` und die Benachrichtigung für den angegebenen S3-Bucket aktiviert

```
$topic = [Amazon.S3.Model.TopicConfiguration] @{  
    Id = "delete-event"  
    Topic = "arn:aws:sns:eu-west-1:123456789012:topic-1"  
    Event = [Amazon.S3.EventType]::ObjectRemovedDelete  
}  
  
Write-S3BucketNotification -BucketName kt-tools -TopicConfiguration $topic
```

Beispiel 2: Dieses Beispiel aktiviert Benachrichtigungen `ObjectCreatedAll` für den angegebenen Bucket und sendet ihn an die Lambda-Funktion.

```
$lambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{  
    Events = "s3:ObjectCreated:*"  
    FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:rdplock"  
    Id = "ObjectCreated-Lambda"  
    Filter = @{  
        S3KeyFilter = @{  
            FilterRules = @(  
                @{Name="Prefix";Value="dada"}  
                @{Name="Suffix";Value=".pem"}  
            )  
        }  
    }  
}  
  
Write-S3BucketNotification -BucketName ssm-editor -LambdaFunctionConfiguration  
$lambdaConfig
```


Beispiel 3: In diesem Beispiel werden zwei verschiedene Lambda-Konfigurationen auf der Grundlage unterschiedlicher Schlüssel-suffixe erstellt und beide in einem einzigen Befehl konfiguriert.

```
#Lambda Config 1

$firstLambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
    Events = "s3:ObjectCreated:*"
    FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:verifynet"
    Id = "ObjectCreated-dada-ps1"
    Filter = @{
        S3KeyFilter = @{
            FilterRules = @(
                @{Name="Prefix";Value="dada"}
                @{Name="Suffix";Value=".ps1"}
            )
        }
    }
}

#Lambda Config 2

$secondLambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
    Events = [Amazon.S3.EventType]::ObjectCreatedAll
    FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:verifyssm"
    Id = "ObjectCreated-dada-json"
    Filter = @{
        S3KeyFilter = @{
            FilterRules = @(
                @{Name="Prefix";Value="dada"}
                @{Name="Suffix";Value=".json"}
            )
        }
    }
}

Write-S3BucketNotification -BucketName ssm-editor -LambdaFunctionConfiguration
    $firstLambdaConfig,$secondLambdaConfig
```

- Einzelheiten finden Sie unter API [PutBucketNotification](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketNotificationConfiguration** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketNotificationConfiguration`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [S3-Ereignisbenachrichtigungen verarbeiten](#)
- [Senden Sie Ereignisbenachrichtigungen an EventBridge](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to enable notifications for an Amazon Simple
/// Storage Service (Amazon S3) bucket.
/// </summary>
public class EnableNotifications
{
    public static async Task Main()
    {
```

```
const string bucketName = "amzn-s3-demo-bucket1";
const string snsTopic = "arn:aws:sns:us-east-2:0123456789ab:bucket-
notify";

const string sqsQueue = "arn:aws:sqs:us-
east-2:0123456789ab:Example_Queue";

IAmazonS3 client = new AmazonS3Client(Amazon.RegionEndpoint.USEast2);
await EnableNotificationAsync(client, bucketName, snsTopic,
sqsQueue);
}

/// <summary>
/// This method makes the call to the PutBucketNotificationAsync method.
/// </summary>
/// <param name="client">An initialized Amazon S3 client used to call
/// the PutBucketNotificationAsync method.</param>
/// <param name="bucketName">The name of the bucket for which
/// notifications will be turned on.</param>
/// <param name="snsTopic">The ARN for the Amazon Simple Notification
/// Service (Amazon SNS) topic associated with the S3 bucket.</param>
/// <param name="sqsQueue">The ARN of the Amazon Simple Queue Service
/// (Amazon SQS) queue to which notifications will be pushed.</param>
public static async Task EnableNotificationAsync(
    IAmazonS3 client,
    string bucketName,
    string snsTopic,
    string sqsQueue)
{
    try
    {
        // The bucket for which we are setting up notifications.
        var request = new PutBucketNotificationRequest()
        {
            BucketName = bucketName,
        };

        // Defines the topic to use when sending a notification.
        var topicConfig = new TopicConfiguration()
        {
            Events = new List<EventType> { EventType.ObjectCreatedCopy },
            Topic = snsTopic,
        };
        request.TopicConfigurations = new List<TopicConfiguration>
        {
```

```

        topicConfig,
    };
    request.QueueConfigurations = new List<QueueConfiguration>
    {
        new QueueConfiguration()
        {
            Events = new List<EventType>
            { EventType.ObjectCreatedPut },
            Queue = sqsQueue,
        },
    };

    // Now apply the notification settings to the bucket.
    PutBucketNotificationResponse response = await
client.PutBucketNotificationAsync(request);
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}
}

```

- API-Einheiten finden Sie [PutBucketNotificationConfiguration](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um die angegebenen Benachrichtigungen für einen Bucket zu aktivieren

Im folgenden `put-bucket-notification-configuration` Beispiel wird eine Benachrichtigungskonfiguration auf einen Bucket mit dem Namen `angewendetmy-bucket`. Die Datei `notification.json` ist ein JSON-Dokument im aktuellen Ordner, das ein SNS-Thema und einen zu überwachten Ereignistyp angibt.

```

aws s3api put-bucket-notification-configuration \
  --bucket my-bucket \
  --notification-configuration file://notification.json

```

Inhalt von notification.json:

```
{
  "TopicConfigurations": [
    {
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-
topic",
      "Events": [
        "s3:ObjectCreated:*"
      ]
    }
  ]
}
```

Dem SNS Thema muss eine IAM Richtlinie beigefügt sein, die es Amazon S3 ermöglicht, darin zu veröffentlichen.

```
{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:us-west-2:123456789012::s3-notification-
topic",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
        }
      }
    }
  ]
}
```

- API-Einheiten finden Sie [PutBucketNotificationConfiguration](#) unter AWS CLI Befehlsreferenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketPolicy** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketPolicy`.

C++

SDK für C++

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::putBucketPolicy(const Aws::String &bucketName,
                                const Aws::String &policyBody,
                                const Aws::S3::S3ClientConfiguration
                                &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    std::shared_ptr<Aws::StringStream> request_body =
        Aws::MakeShared<Aws::StringStream>("");
    *request_body << policyBody;

    Aws::S3::Model::PutBucketPolicyRequest request;
    request.SetBucket(bucketName);
    request.SetBody(request_body);

    Aws::S3::Model::PutBucketPolicyOutcome outcome =
        s3Client.PutBucketPolicy(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: putBucketPolicy: "
                  << outcome.GetError().GetMessage() << std::endl;
    }
}
```

```

    } else {
        std::cout << "Set the following policy body for the bucket '" <<
            bucketName << "':" << std::endl << std::endl;
        std::cout << policyBody << std::endl;
    }

    return outcome.IsSuccess();
}

//! Build a policy JSON string.
/*!
    \param userArn: Aws user Amazon Resource Name (ARN).
        For more information, see https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_identifiers.html#identifiers-arns.
    \param bucketName: Name of a bucket.
    \return String: Policy as JSON string.
*/

Aws::String getPolicyString(const Aws::String &userArn,
                           const Aws::String &bucketName) {
    return
        "{\n"
        "  \"Version\": \"2012-10-17\", \n"
        "  \"Statement\": [\n"
        "    {\n"
        "      \"Sid\": \"1\", \n"
        "      \"Effect\": \"Allow\", \n"
        "      \"Principal\": {\n"
        "        \"AWS\": \""
        + userArn +
        "\"\n"
        "      }, \n"
        "      \"Action\": [ \"s3:getObject\" ], \n"
        "      \"Resource\": [ \"arn:aws:s3:::"
        + bucketName +
        "\"/*\" ] \n"
        "    } \n"
        "  ] \n"
        "}";
}

```

- API-Einheiten finden Sie [PutBucketPolicy](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

In diesem Beispiel können alle Benutzer alle Objekte abrufen, MyBucket mit Ausnahme der Objekte in MySecretFolder. Es gewährt put auch delete dem Root-Benutzer von AWS Konto1234-5678-9012:

```
aws s3api put-bucket-policy --bucket MyBucket --policy file://policy.json
```

```
policy.json:
```

```
{  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::MyBucket/*"  
    },  
    {  
      "Effect": "Deny",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::MyBucket/MySecretFolder/*"  
    },  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::123456789012:root"  
      },  
      "Action": [  
        "s3:DeleteObject",  
        "s3:PutObject"  
      ],  
      "Resource": "arn:aws:s3:::MyBucket/*"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [PutBucketPolicy](#) in AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutBucketPolicyRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class SetBucketPolicy {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <bucketName> <polFile>

                Where:
                bucketName - The Amazon S3 bucket to set the policy on.
                polFile - A JSON file containing the policy (see the Amazon
                S3 Readme for an example).\s
```

```
        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String bucketName = args[0];
    String polFile = args[1];
    String policyText = getBucketPolicyFromFile(polFile);
    Region region = Region.US_EAST_1;
    S3Client s3 = S3Client.builder()
        .region(region)
        .build();

    setPolicy(s3, bucketName, policyText);
    s3.close();
}

public static void setPolicy(S3Client s3, String bucketName, String
policyText) {
    System.out.println("Setting policy:");
    System.out.println("----");
    System.out.println(policyText);
    System.out.println("----");
    System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

    try {
        PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
            .bucket(bucketName)
            .policy(policyText)
            .build();

        s3.putBucketPolicy(policyReq);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    System.out.println("Done!");
}

// Loads a JSON-formatted policy from a file
```

```
public static String getBucketPolicyFromFile(String policyFile) {

    StringBuilder fileText = new StringBuilder();
    try {
        List<String> lines = Files.readAllLines(Paths.get(policyFile),
StandardCharsets.UTF_8);
        for (String line : lines) {
            fileText.append(line);
        }

    } catch (IOException e) {
        System.out.format("Problem reading file: \"%s\"", policyFile);
        System.out.println(e.getMessage());
    }

    try {
        final JsonParser parser = new
ObjectMapper().getFactory().createParser(fileText.toString());
        while (parser.nextToken() != null) {
        }

    } catch (IOException jpe) {
        jpe.printStackTrace();
    }
    return fileText.toString();
}
}
```

- APIEinzelheiten finden Sie [PutBucketPolicy](#) unter AWS SDK for Java 2.x APIReferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Fügen Sie die Richtlinie hinzu.

```
import { PutBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new PutBucketPolicyCommand({
    Policy: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Sid: "AllowGetObject",
          // Allow this particular user to call GetObject on any object in this
bucket.
          Effect: "Allow",
          Principal: {
            AWS: "arn:aws:iam::ACCOUNT-ID:user/USERNAME",
          },
          Action: "s3:GetObject",
          Resource: "arn:aws:s3:::BUCKET-NAME/*",
        },
      ],
    }),
    // Apply the preceding policy to this bucket.
    Bucket: "BUCKET-NAME",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [PutBucketPolicy](#) in AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
        Boto3
            that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def put_policy(self, policy):
        """
        Apply a security policy to the bucket. Policies control users' ability
        to perform specific actions, such as listing the objects in the bucket.

        :param policy: The policy to apply to the bucket.
        """
        try:
            self.bucket.Policy().put(Policy=json.dumps(policy))
            logger.info("Put policy %s for bucket '%s'.", policy,
self.bucket.name)
        except ClientError:
            logger.exception("Couldn't apply policy to bucket '%s'.",
self.bucket.name)
            raise
```

- API-Einheiten finden Sie [PutBucketPolicy](#) unter AWS SDK für Python (Boto3) API - Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object
  # configured with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  # Sets a policy on a bucket.
  #
  def set_policy(policy)
    @bucket_policy.put(policy: policy)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't set the policy for #{@bucket_policy.bucket.name}. Here's why:
#{e.message}"
    false
  end
end
```

- API-Einheiten finden Sie [PutBucketPolicy](#) unter AWS SDK for Ruby API Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketReplication** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketReplication`.

CLI

AWS CLI

Um die Replikation für einen S3-Bucket zu konfigurieren

Im folgenden `put-bucket-replication` Beispiel wird eine Replikationskonfiguration auf den angegebenen S3-Bucket angewendet.

```
aws s3api put-bucket-replication \  
  --bucket AWSDOC-EXAMPLE-BUCKET1 \  
  --replication-configuration file://replication.json
```

Inhalt von `replication.json`:

```
{  
  "Role": "arn:aws:iam::123456789012:role/s3-replication-role",  
  "Rules": [  
    {  
      "Status": "Enabled",  
      "Priority": 1,  
      "DeleteMarkerReplication": { "Status": "Disabled" },  
      "Filter" : { "Prefix": "" },  
      "Destination": {  
        "Bucket": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2"  
      }  
    }  
  ]  
}
```

Für den Ziel-Bucket muss die Versionierung aktiviert sein. Die angegebene Rolle muss über Schreibberechtigungen in den Ziel-Bucket verfügen und über eine Vertrauensbeziehung verfügen, die es Amazon S3 ermöglicht, die Rolle zu übernehmen.

Beispiel für eine Rollenberechtigungsrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetReplicationConfiguration",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
      ],
      "Resource": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2/*"
    }
  ]
}
```

Beispiel für eine Vertrauensbeziehungsrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
```



```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "s3.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dies ist der Thementitel](#) im Amazon Simple Storage Service Console-Benutzerhandbuch.

- API-Einheiten finden Sie [PutBucketReplication](#) unter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: In diesem Beispiel wird eine Replikationskonfiguration mit einer einzigen Regel eingerichtet, die die Replikation aller neuen Objekte, die mit dem Schlüsselnamenpräfix "TaxDocs" im Bucket 'examplebucket' erstellt wurden, in den Bucket 'exampletargetbucket' ermöglicht.

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Enabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
  BucketName = "examplebucket"
  Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
  Configuration_Rule = $rule1
}

Write-S3BucketReplication @params
```

Beispiel 2: In diesem Beispiel wird eine Replikationskonfiguration mit mehreren Regeln festgelegt, die die Replikation aller neuen Objekte, die entweder mit dem Schlüsselnamenpräfix "" oder "" erstellt wurden, in den Bucket 'exampletargetbucket' ermöglichen. TaxDocs OtherDocs Die Schlüsselpräfixe dürfen sich nicht überschneiden.

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Enabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$rule2 = New-Object Amazon.S3.Model.ReplicationRule
$rule2.ID = "Rule-2"
$rule2.Status = "Enabled"
$rule2.Prefix = "OtherDocs"
$rule2.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1,$rule2
}

Write-S3BucketReplication @params
```

Beispiel 3: In diesem Beispiel wird die Replikationskonfiguration für den angegebenen Bucket aktualisiert, um die Regel zu deaktivieren, die die Replikation von Objekten mit dem Schlüsselnamenpräfix "TaxDocs" in den Bucket 'exampletargetbucket' steuert.

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Disabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1
}
```

```
Write-S3BucketReplication @params
```

- Einzelheiten finden Sie unter API [PutBucketReplication](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketRequestPayment** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketRequestPayment`.

CLI

AWS CLI

Beispiel 1: Um die `requester pays`-Konfiguration für einen Bucket zu aktivieren

Das folgende `put-bucket-request-payment` Beispiel aktiviert für den angegebenen Bucket. `requester pays`

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"Requester"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um die Konfiguration `requester pays` für einen Bucket zu deaktivieren

Das folgende `put-bucket-request-payment` Beispiel deaktiviert die Option für den angegebenen Bucket. `requester pays`

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einzelheiten finden Sie unter [PutBucketRequestPayment](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Aktualisiert die Konfiguration der Zahlungsanfrage für den Bucket mit dem Namen „mybucket“, sodass der Person, die Downloads aus dem Bucket anfordert, der Download in Rechnung gestellt wird. Standardmäßig zahlt der Bucket-Besitzer für Downloads. Verwenden Sie 'BucketOwner' für den Parameter RequestPaymentConfiguration _Payer, um die Zahlung für die Anfrage wieder auf die Standardwerte zurückzusetzen.

```
Write-S3BucketRequestPayment -BucketName mybucket -  
RequestPaymentConfiguration_Payer Requester
```

- API-Einheiten finden Sie unter [PutBucketRequestPayment](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketTagging** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketTagging`.

CLI

AWS CLI

Der folgende Befehl wendet eine Tagging-Konfiguration auf einen Bucket mit dem Namen my-bucket an:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

Die Datei `tagging.json` ist ein JSON-Dokument im aktuellen Ordner, das Tags angibt:

```
{  
  "TagSet": [  
    {  
      "Key": "organization",  
      "Value": "marketing"  
    }  
  ]  
}
```

```
]
}
```

Oder wenden Sie eine Tagging-Konfiguration `my-bucket` direkt von der Befehlszeile aus an:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
'TagSet=[{Key=organization, Value=marketing}]'
```

- API-Einheiten finden Sie unter [PutBucketTagging](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl wendet zwei Tags auf einen Bucket mit dem Namen `ancloudtrail-test-2018`: ein Tag mit dem Schlüssel `Stage` und dem Wert `Test` und ein Tag mit dem Schlüssel `Environment` und dem Wert `Alpha`. Führen Sie den Befehl aus, um zu überprüfen, ob die Tags dem Bucket hinzugefügt wurden `Get-S3BucketTagging -BucketName bucket_name`. Die Ergebnisse sollten die Tags enthalten, die Sie im ersten Befehl auf den Bucket angewendet haben. Beachten Sie, dass `Write-S3BucketTagging` dadurch der gesamte vorhandene Tagsatz in einem Bucket überschrieben wird. Um einzelne Tags hinzuzufügen oder zu löschen, führen Sie die API Cmdlets `Resource Groups` und `Tagging` aus. `Add-RGResourceTag` `Remove-RGResourceTag` Verwenden Sie alternativ den Tag-Editor in der AWS Management Console zur Verwaltung von S3-Bucket-Tags.

```
Write-S3BucketTagging -BucketName cloudtrail-test-2018 -TagSet @( @{ Key="Stage";
Value="Test" }, @{ Key="Environment"; Value="Alpha" } )
```

Beispiel 2: Mit diesem Befehl wird ein Bucket mit dem Namen `ancloudtrail-test-2018` das `Write-S3BucketTagging` Cmdlet übergeben. Er wendet die Tags `Stage:Production` und `Department:Finance` auf den Bucket an. Beachten Sie, dass dadurch der gesamte vorhandene Tagsatz in einem `Write-S3BucketTagging` Bucket überschrieben wird.

```
Get-S3Bucket -BucketName cloudtrail-test-2018 | Write-S3BucketTagging
-TagSet @( @{ Key="Stage"; Value="Production" }, @{ Key="Department";
Value="Finance" } )
```

- API-Einheiten finden Sie unter [PutBucketTagging](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketVersioning** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutBucketVersioning`.

CLI

AWS CLI

Der folgende Befehl aktiviert die Versionierung für einen Bucket mit dem Namen `my-bucket`:

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration Status=Enabled
```

Der folgende Befehl aktiviert die Versionierung und verwendet einen MFA-Code

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration Status=Enabled --mfa "SERIAL 123456"
```

- API-Einheiten finden Sie unter [PutBucketVersioning](#) AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl aktiviert die Versionierung für den angegebenen S3-Bucket.

```
Write-S3BucketVersioning -BucketName 's3testbucket' -VersioningConfig_Status Enabled
```

- API-Einheiten finden Sie unter [PutBucketVersioning](#) AWS Tools for PowerShell Cmdlet-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutBucketWebsite** mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen, wie man es benutztPutBucketWebsite.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// Put the website configuration.
PutBucketWebsiteRequest putRequest = new
PutBucketWebsiteRequest()
{
    BucketName = bucketName,
    WebsiteConfiguration = new WebsiteConfiguration()
    {
        IndexDocumentSuffix = indexDocumentSuffix,
        ErrorDocument = errorDocument,
    },
};
PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);
```

- APIEinzelheiten finden Sie [PutBucketWebsite](#)unter AWS SDK for .NET APIReferenz.

C++

SDKfür C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::putWebsiteConfig(const Aws::String &bucketName,
                                   const Aws::String &indexPath, const Aws::String
                                   &errorPage,
                                   const Aws::S3::S3ClientConfiguration
                                   &clientConfig) {
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::IndexDocument indexDocument;
    indexDocument.SetSuffix(indexPath);

    Aws::S3::Model::ErrorDocument errorDocument;
    errorDocument.SetKey(errorPage);

    Aws::S3::Model::WebsiteConfiguration websiteConfiguration;
    websiteConfiguration.SetIndexDocument(indexDocument);
    websiteConfiguration.SetErrorDocument(errorDocument);

    Aws::S3::Model::PutBucketWebsiteRequest request;
    request.SetBucket(bucketName);
    request.SetWebsiteConfiguration(websiteConfiguration);

    Aws::S3::Model::PutBucketWebsiteOutcome outcome =
        client.PutBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: PutBucketWebsite: "
                  << outcome.GetError().GetMessage() << std::endl;
    } else {
        std::cout << "Success: Set website configuration for bucket '"
                  << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- API-Einheiten finden Sie [PutBucketWebsite](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Das wendet eine statische Website-Konfiguration auf einen Bucket mit dem Namen `amy-bucket`:

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://website.json
```

Die Datei `website.json` ist ein JSON Dokument im aktuellen Ordner, das Index- und Fehlerseiten für die Website angibt:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

- API-Einheiten finden Sie [PutBucketWebsite](#) unter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.IndexDocument;
import software.amazon.awssdk.services.s3.model.PutBucketWebsiteRequest;
import software.amazon.awssdk.services.s3.model.WebsiteConfiguration;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class SetWebsiteConfiguration {
    public static void main(String[] args) {
        final String usage = ""

            Usage:    <bucketName> [indexdoc]\s

            Where:
                bucketName    - The Amazon S3 bucket to set the website
configuration on.\s
                indexdoc    - The index document, ex. 'index.html'
                            If not specified, 'index.html' will be set.

            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String indexDoc = "index.html";
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        setWebsiteConfig(s3, bucketName, indexDoc);
        s3.close();
    }

    public static void setWebsiteConfig(S3Client s3, String bucketName, String
indexDoc) {
        try {
            WebsiteConfiguration websiteConfig = WebsiteConfiguration.builder()
```

```
.indexDocument(IndexDocument.builder().suffix(indexDoc).build())
    .build();

    PutBucketWebsiteRequest pubWebsiteReq =
PutBucketWebsiteRequest.builder()
    .bucket(bucketName)
    .websiteConfiguration(websiteConfig)
    .build();

    s3.putBucketWebsite(pubWebsiteReq);
    System.out.println("The call was successful");

} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

- API-Einheiten finden Sie [PutBucketWebsite](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Legen Sie die Website-Konfiguration fest.

```
import { PutBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Set up a bucket as a static website.
// The bucket needs to be publicly accessible.
export const main = async () => {
```

```
const command = new PutBucketWebsiteCommand({
  Bucket: "test-bucket",
  WebsiteConfiguration: {
    ErrorDocument: {
      // The object key name to use when a 4XX class error occurs.
      Key: "error.html",
    },
    IndexDocument: {
      // A suffix that is appended to a request that is for a directory.
      Suffix: "index.html",
    },
  },
});

try {
  const response = await client.send(command);
  console.log(response);
} catch (err) {
  console.error(err);
}
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [PutBucketWebsite](#) in AWS SDK for JavaScript API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl aktiviert das Website-Hosting für den angegebenen Bucket mit dem Indextokument als 'index.html' und dem Fehlerdokument als 'error.html'.

```
Write-S3BucketWebsite -BucketName 's3testbucket' -
WebsiteConfiguration_IndexDocumentSuffix 'index.html' -
WebsiteConfiguration_ErrorDocument 'error.html'
```

- Einzelheiten finden Sie unter API [PutBucketWebsite](#) AWS Tools for PowerShell Cmdlet-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket website actions.
class BucketWebsiteWrapper
  attr_reader :bucket_website

  # @param bucket_website [Aws::S3::BucketWebsite] A bucket website object
  # configured with an existing bucket.
  def initialize(bucket_website)
    @bucket_website = bucket_website
  end

  # Sets a bucket as a static website.
  #
  # @param index_document [String] The name of the index document for the
  # website.
  # @param error_document [String] The name of the error document to show for 4XX
  # errors.
  # @return [Boolean] True when the bucket is configured as a website; otherwise,
  # false.
  def set_website(index_document, error_document)
    @bucket_website.put(
      website_configuration: {
        index_document: { suffix: index_document },
        error_document: { key: error_document }
      }
    )
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't configure #{@bucket_website.bucket.name} as a website. Here's
  why: #{e.message}"
    false
  end
end
```

```
end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  index_document = "index.html"
  error_document = "404.html"

  wrapper = BucketWebsiteWrapper.new(Aws::S3::BucketWebsite.new(bucket_name))
  return unless wrapper.set_website(index_document, error_document)

  puts "Successfully configured bucket #{bucket_name} as a static website."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- API-Einheiten finden Sie [PutBucketWebsite](#) unter AWS SDK for Ruby API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutObject** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutObject`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erlernen Sie die Grundlagen](#)
- [Verfolgen Sie Uploads und Downloads](#)
- [Arbeiten Sie mit Amazon S3 S3-Objektintegrität](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Shows how to upload a file from the local computer to an Amazon S3
/// bucket.
/// </summary>
/// <param name="client">An initialized Amazon S3 client object.</param>
/// <param name="bucketName">The Amazon S3 bucket to which the object
/// will be uploaded.</param>
/// <param name="objectName">The object to upload.</param>
/// <param name="filePath">The path, including file name, of the object
/// on the local computer to upload.</param>
/// <returns>A boolean value indicating the success or failure of the
/// upload procedure.</returns>
public static async Task<bool> UploadFileAsync(
    IAmazonS3 client,
    string bucketName,
    string objectName,
    string filePath)
{
    var request = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = objectName,
        FilePath = filePath,
    };

    var response = await client.PutObjectAsync(request);
    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
    {
        Console.WriteLine($"Successfully uploaded {objectName} to
{bucketName}.");
        return true;
    }
}
```

```
    }
    else
    {
        Console.WriteLine($"Could not upload {objectName} to
{bucketName}.");
        return false;
    }
}
```

Laden Sie ein Objekt mit serverseitiger Verschlüsselung hoch.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to upload an object to an Amazon Simple Storage
/// Service (Amazon S3) bucket with server-side encryption enabled.
/// </summary>
public class ServerSideEncryption
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";
        string keyName = "samplefile.txt";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2.
        IAmazonS3 client = new AmazonS3Client();

        await WritingAnObjectAsync(client, bucketName, keyName);
    }

    /// <summary>
    /// Upload a sample object include a setting for encryption.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
    /// to upload a file and apply server-side encryption.</param>
}
```



```
    /// <param name="bucketName">The name of the Amazon S3 bucket where the
    /// encrypted object will reside.</param>
    /// <param name="keyName">The name for the object that you want to
    /// create in the supplied bucket.</param>
    public static async Task WritingAnObjectAsync(IAmazonS3 client, string
bucketName, string keyName)
    {
        try
        {
            var putRequest = new PutObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
                ContentBody = "sample text",
                ServerSideEncryptionMethod =
ServerSideEncryptionMethod.AES256,
            };

            var putResponse = await client.PutObjectAsync(putRequest);

            // Determine the encryption state of an object.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
            {
                BucketName = bucketName,
                Key = keyName,
            };
            GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);
            ServerSideEncryptionMethod objectEncryption =
response.ServerSideEncryptionMethod;

            Console.WriteLine($"Encryption method used: {0}",
objectEncryption.ToString());
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error: '{ex.Message}' when writing an
object");
        }
    }
}
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK for .NET API-Referenz.

Bash

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file to.
#     $2 - The path and file name of the local file to copy to the bucket.
#     $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_file_to_bucket() {
    local response bucket_name source_file destination_file_name
    bucket_name=$1
    source_file=$2
    destination_file_name=$3
```

```

response=$(aws s3api put-object \
  --bucket "$bucket_name" \
  --body "$source_file" \
  --key "$destination_file_name")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
  errecho "ERROR: AWS reports put-object operation failed.\n$response"
  return 1
fi
}

```

- API-Einheiten finden Sie [PutObject](#) unter AWS CLI Befehlsreferenz.

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

bool AwsDoc::S3::putObject(const Aws::String &bucketName,
                          const Aws::String &fileName,
                          const Aws::S3::S3ClientConfiguration &clientConfig) {
  Aws::S3::S3Client s3Client(clientConfig);

  Aws::S3::Model::PutObjectRequest request;
  request.SetBucket(bucketName);
  //We are using the name of the file as the key for the object in the bucket.
  //However, this is just a string and can be set according to your retrieval
  needs.
  request.SetKey(fileName);

  std::shared_ptr<Aws::IOStream> inputData =
    Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
                                  fileName.c_str(),

```

```
std::ios_base::in |
std::ios_base::binary);

if (!*inputData) {
    std::cerr << "Error unable to read file " << fileName << std::endl;
    return false;
}

request.SetBody(inputData);

Aws::S3::Model::PutObjectOutcome outcome =
    s3Client.PutObject(request);

if (!outcome.IsSuccess()) {
    std::cerr << "Error: putObject: " <<
        outcome.GetError().GetMessage() << std::endl;
} else {
    std::cout << "Added object '" << fileName << "' to bucket '"
        << bucketName << "'.";
}

return outcome.IsSuccess();
}
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Im folgenden Beispiel wird der `put-object` Befehl verwendet, um ein Objekt auf Amazon S3 hochzuladen:

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --
body my_images.tar.bz2
```

Das folgende Beispiel zeigt den Upload einer Videodatei (Die Videodatei wird mithilfe der Windows-Dateisystemsyntax spezifiziert.):

```
aws s3api put-object --bucket text-content --key dir-1/big-video-file.mp4 --body  
e:\media\videos\f-sharp-3-data-services.mp4
```

Weitere Informationen zum Hochladen von Objekten finden Sie unter [Hochladen von Objekten im Amazon S3 Developer Guide](#).

- API-Einheiten finden Sie unter [PutObject](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu [GitHub](#). Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Platzieren Sie ein Objekt mithilfe des Low-Levels API in einem Bucket.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)  
// actions  
// used in the examples.  
// It contains S3Client, an Amazon S3 service client that is used to perform  
// bucket  
// and object actions.  
type BucketBasics struct {  
    S3Client *s3.Client  
}  
  
// UploadFile reads from a file and puts the data into an object in a bucket.  
func (basics BucketBasics) UploadFile(bucketName string, objectKey string,  
    fileName string) error {  
    file, err := os.Open(fileName)  
    if err != nil {  
        log.Printf("Couldn't open file %v to upload. Here's why: %v\n", fileName, err)  
    } else {  
        defer file.Close()
```

```

_, err = basics.S3Client.PutObject(context.TODO(), &s3.PutObjectInput{
    Bucket: aws.String(bucketName),
    Key:    aws.String(objectKey),
    Body:   file,
})
if err != nil {
    log.Printf("Couldn't upload file %v to %v:%v. Here's why: %v\n",
        fileName, bucketName, objectKey, err)
}
}
return err
}

```

Laden Sie ein Objekt mithilfe eines Transfer-Managers in einen Bucket hoch.

```

// S3Actions wraps S3 service actions.
type S3Actions struct {
    S3Client *s3.Client
    S3Manager *manager.Uploader
}

// UploadObject uses the S3 upload manager to upload an object to a bucket.
func (actor S3Actions) UploadObject(ctx context.Context, bucket string, key
string, contents string) (string, error) {
    var outKey string
    input := &s3.PutObjectInput{
        Bucket:          aws.String(bucket),
        Key:             aws.String(key),
        Body:            bytes.NewReader([]byte(contents)),
        ChecksumAlgorithm: types.ChecksumAlgorithmSha256,
    }
    output, err := actor.S3Manager.Upload(ctx, input)
    if err != nil {
        var noBucket *types.NoSuchBucket
        if errors.As(err, &noBucket) {
            log.Printf("Bucket %s does not exist.\n", bucket)
            err = noBucket
        }
    }
}

```

```
} else {
    err := s3.NewObjectExistsWaiter(actor.S3Client).Wait(ctx, &s3.HeadObjectInput{
        Bucket: aws.String(bucket),
        Key:     aws.String(key),
    }, time.Minute)
    if err != nil {
        log.Printf("Failed attempt to wait for object %s to exist in %s.\n", key,
            bucket)
    } else {
        outKey = *output.Key
    }
}
return outKey, err
}
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Laden Sie eine Datei mit einem [S3Client](#) in einen Bucket hoch.

```
/**
 * Uploads a local file to an AWS S3 bucket asynchronously.
 *
 * @param bucketName the name of the S3 bucket to upload the file to
 * @param key the key (object name) to use for the uploaded file
 * @param objectPath the local file path of the file to be uploaded
 * @return a {@link CompletableFuture} that completes with the {@link
 * PutObjectResponse} when the upload is successful, or throws a {@link
 * RuntimeException} if the upload fails
 */
```

```
public CompletableFuture<PutObjectResponse> uploadLocalFileAsync(String
bucketName, String key, String objectPath) {
    PutObjectRequest objectRequest = PutObjectRequest.builder()
        .bucket(bucketName)
        .key(key)
        .build();

    CompletableFuture<PutObjectResponse> response =
getAsyncClient().putObject(objectRequest,
AsyncRequestBody.fromFile(Paths.get(objectPath)));
    return response.whenComplete((resp, ex) -> {
        if (ex != null) {
            throw new RuntimeException("Failed to upload file", ex);
        }
    });
}
```

Verwenden Sie ein [S3 TransferManager](#), um [eine Datei in einen Bucket hochzuladen](#). Sehen Sie sich die [vollständige Datei](#) an und [testen](#) Sie sie.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileUpload;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

public String uploadFile(S3TransferManager transferManager, String
bucketName,
                        String key, URI filePathURI) {
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b.bucket(bucketName).key(key))
        .source(Paths.get(filePathURI))
        .build();

    FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
```



```
CompletedFileUpload uploadResult = fileUpload.completionFuture().join();
return uploadResult.response().eTag();
}
```

Laden Sie ein Objekt mit einem [S3Client](#) in einen Bucket hoch und legen Sie Tags fest.

```
public static void putS3ObjectTags(S3Client s3, String bucketName, String
objectKey, String objectPath) {
    try {
        Tag tag1 = Tag.builder()
            .key("Tag 1")
            .value("This is tag 1")
            .build();

        Tag tag2 = Tag.builder()
            .key("Tag 2")
            .value("This is tag 2")
            .build();

        List<Tag> tags = new ArrayList<>();
        tags.add(tag1);
        tags.add(tag2);

        Tagging allTags = Tagging.builder()
            .tagSet(tags)
            .build();

        PutObjectRequest putOb = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .tagging(allTags)
            .build();

        s3.putObject(putOb,
RequestBody.fromBytes(getObjectFile(objectPath)));

    } catch (S3Exception e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
```

```
public static void updateObjectTags(S3Client s3, String bucketName, String
objectKey) {
    try {
        GetObjectTaggingRequest taggingRequest =
GetObjectTaggingRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .build();

        GetObjectTaggingResponse getTaggingRes =
s3.getObjectTagging(taggingRequest);
        List<Tag> obTags = getTaggingRes.tagSet();
        for (Tag sinTag : obTags) {
            System.out.println("The tag key is: " + sinTag.key());
            System.out.println("The tag value is: " + sinTag.value());
        }

        // Replace the object's tags with two new tags.
        Tag tag3 = Tag.builder()
        .key("Tag 3")
        .value("This is tag 3")
        .build();

        Tag tag4 = Tag.builder()
        .key("Tag 4")
        .value("This is tag 4")
        .build();

        List<Tag> tags = new ArrayList<>();
        tags.add(tag3);
        tags.add(tag4);

        Tagging updatedTags = Tagging.builder()
        .tagSet(tags)
        .build();

        PutObjectTaggingRequest taggingRequest1 =
PutObjectTaggingRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .tagging(updatedTags)
        .build();
    }
}
```

```
s3.putObjectTagging(taggingRequest1);
GetObjectTaggingResponse getTaggingRes2 =
s3.getObjectTagging(taggingRequest);
List<Tag> modTags = getTaggingRes2.tagSet();
for (Tag sinTag : modTags) {
    System.out.println("The tag key is: " + sinTag.key());
    System.out.println("The tag value is: " + sinTag.value());
}

} catch (S3Exception e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
}

// Return a byte array.
private static byte[] getObjectFile(String filePath) {
    FileInputStream fileInputStream = null;
    byte[] byteArray = null;

    try {
        File file = new File(filePath);
        byteArray = new byte[(int) file.length()];
        fileInputStream = new FileInputStream(file);
        fileInputStream.read(byteArray);

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fileInputStream != null) {
            try {
                fileInputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    return byteArray;
}
}
```

Laden Sie ein Objekt mit einem [S3Client](#) in einen Bucket hoch und legen Sie Metadaten fest.

```
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class PutObjectMetadata {
    public static void main(String[] args) {
        final String USAGE = ""

            Usage:
                <bucketName> <objectKey> <objectPath>\s

            Where:
                bucketName - The Amazon S3 bucket to upload an object into.
                objectKey - The object to upload (for example, book.pdf).
                objectPath - The path where the file is located (for example,
C:/AWS/book2.pdf).\s
            """;

        if (args.length != 3) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String bucketName = args[0];
        String objectKey = args[1];
        String objectPath = args[2];
        System.out.println("Putting object " + objectKey + " into bucket " +
bucketName);
    }
}
```

```
        System.out.println("  in bucket: " + bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        putS3Object(s3, bucketName, objectKey, objectPath);
        s3.close();
    }

    // This example uses RequestBody.fromFile to avoid loading the whole file
    into
    // memory.
    public static void putS3Object(S3Client s3, String bucketName, String
    objectKey, String objectPath) {
        try {
            Map<String, String> metadata = new HashMap<>();
            metadata.put("author", "Mary Doe");
            metadata.put("version", "1.0.0.0");

            PutObjectRequest putOb = PutObjectRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .metadata(metadata)
                .build();

            s3.putObject(putOb, RequestBody.fromFile(new File(objectPath)));
            System.out.println("Successfully placed " + objectKey + " into bucket
            " + bucketName);

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

Laden Sie ein Objekt mit einem [S3Client](#) in einen Bucket hoch und legen Sie einen Wert für die Objektaufbewahrung fest.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
```

```
import software.amazon.awssdk.services.s3.model.PutObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.time.Instant;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.ZoneOffset;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class PutObjectRetention {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <key> <bucketName>\s

            Where:
                key - The name of the object (for example, book.pdf).\s
                bucketName - The Amazon S3 bucket name that contains the
object (for example, bucket1).\s
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String key = args[0];
        String bucketName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        setRentionPeriod(s3, key, bucketName);
    }
}
```

```
s3.close();
}

public static void setRetentionPeriod(S3Client s3, String key, String bucket) {
    try {
        LocalDate localDate = LocalDate.parse("2020-07-17");
        LocalDateTime localDateTime = localDate.atStartOfDay();
        Instant instant = localDateTime.toInstant(ZoneOffset.UTC);

        ObjectLockRetention lockRetention = ObjectLockRetention.builder()
            .mode("COMPLIANCE")
            .retainUntilDate(instant)
            .build();

        PutObjectRetentionRequest retentionRequest =
PutObjectRetentionRequest.builder()
            .bucket(bucket)
            .key(key)
            .bypassGovernanceRetention(true)
            .retention(lockRetention)
            .build();

        // To set Retention on an object, the Amazon S3 bucket must support
object
        // locking, otherwise an exception is thrown.
        s3.putObjectRetention(retentionRequest);
        System.out.print("An object retention configuration was successfully
placed on the object");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- APIEinzelheiten finden Sie [PutObject](#) unter AWS SDK for Java 2.x APIReferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Laden Sie das Objekt hoch.

```
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new PutObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden für Entwickler](#).
- API-Einheiten finden Sie [PutObject](#) in AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
suspend fun putS3Object(
    bucketName: String,
    objectKey: String,
    objectPath: String,
) {
    val metadataVal = mutableMapOf<String, String>()
    metadataVal["myVal"] = "test"

    val request =
        PutObjectRequest {
            bucket = bucketName
            key = objectKey
            metadata = metadataVal
            body = File(objectPath).asByteStream()
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.putObject(request)
        println("Tag information is ${response.eTag}")
    }
}
```

- API Einzelheiten finden Sie [PutObject](#) unter AWS SDK als API Kotlin-Referenz.

PHP

SDK für PHP

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Laden Sie ein Objekt in einen Bucket hoch.

```
$s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

$fileName = __DIR__ . "/local-file-" . uniqid();
try {
    $this->s3client->putObject([
        'Bucket' => $this->bucketName,
        'Key' => $fileName,
        'SourceFile' => __DIR__ . '/testfile.txt'
    ]);
    echo "Uploaded $fileName to $this->bucketName.\n";
} catch (Exception $exception) {
    echo "Failed to upload $fileName with error: " . $exception-
    >getMessage();
    exit("Please fix error with file upload before continuing.");
}
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK for PHP API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl lädt die einzelne Datei "local-sample.txt" auf Amazon S3 hoch und erstellt ein Objekt mit dem Schlüssel "sample.txt" im Bucket „test-files“.

```
Write-S3Object -BucketName test-files -Key "sample.txt" -File .\local-sample.txt
```

Beispiel 2: Dieser Befehl lädt die einzelne Datei "sample.txt" auf Amazon S3 hoch und erstellt ein Objekt mit dem Schlüssel "sample.txt" im Bucket „test-files“. Wenn der -Key-Parameter nicht angegeben wird, wird der Dateiname als S3-Objektschlüssel verwendet.

```
Write-S3Object -BucketName test-files -File .\sample.txt
```

Beispiel 3: Dieser Befehl lädt die einzelne Datei "local-sample.txt" auf Amazon S3 hoch und erstellt ein Objekt mit dem Schlüssel "prefix/to/sample.txt" im Bucket „test-files“.

```
Write-S3Object -BucketName test-files -Key "prefix/to/sample.txt" -File .\local-sample.txt
```

Beispiel 4: Dieser Befehl lädt alle Dateien im Unterverzeichnis „Scripts“ in den Bucket „test-files“ hoch und wendet das gemeinsame key prefix "SampleScripts" auf jedes Objekt an. Jede hochgeladene Datei hat den Schlüssel "SampleScripts/filename", wobei 'Dateiname' variiert.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
```

Beispiel 5: Dieser Befehl lädt alle *.ps1-Dateien im lokalen Verzeichnis „Scripts“ in den Bucket „test-files“ hoch und wendet das gemeinsame key prefix "" SampleScripts auf jedes Objekt an. Jede hochgeladene Datei hat den Schlüssel "SampleScripts/filename.ps1", wobei 'Dateiname' variiert.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\  
-SearchPattern *.ps1
```

Beispiel 6: Dieser Befehl erstellt ein neues S3-Objekt, das die angegebene Inhaltszeichenfolge mit dem Schlüssel 'sample.txt' enthält.

```
Write-S3Object -BucketName test-files -Key "sample.txt" -Content "object  
contents"
```

Beispiel 7: Dieser Befehl lädt die angegebene Datei hoch (der Dateiname wird als Schlüssel verwendet) und wendet die angegebenen Tags auf das neue Objekt an.

```
Write-S3Object -BucketName test-files -File "sample.txt" -TagSet  
@{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

Beispiel 8: Dieser Befehl lädt den angegebenen Ordner rekursiv hoch und wendet die angegebenen Tags auf alle neuen Objekte an.

```
Write-S3Object -BucketName test-files -Folder . -KeyPrefix "TaggedFiles" -Recurse
-TagSet @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

- API-Einheiten finden Sie unter [PutObject](#) AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
                               that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def put(self, data):
        """
        Upload data to the object.

        :param data: The data to upload. This can either be bytes or a string.
        When this
                               argument is a string, it is interpreted as a file name,
        which is
                               opened in read bytes mode.
        """
```

```
    put_data = data
    if isinstance(data, str):
        try:
            put_data = open(data, "rb")
        except IOError:
            logger.exception("Expected file name or binary data, got '%s'.",
data)
            raise

    try:
        self.object.put(Body=put_data)
        self.object.wait_until_exists()
        logger.info(
            "Put object '%s' to bucket '%s'.",
            self.object.key,
            self.object.bucket_name,
        )
    except ClientError:
        logger.exception(
            "Couldn't put object '%s' to bucket '%s'.",
            self.object.key,
            self.object.bucket_name,
        )
        raise
    finally:
        if getattr(put_data, "close", None):
            put_data.close()
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK für Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Laden Sie eine Datei mit einem verwalteten Uploader (`Object.upload_file`) hoch.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Uploads a file to an Amazon S3 object by using a managed uploader.
  #
  # @param file_path [String] The path to the file to upload.
  # @return [Boolean] True when the file is uploaded; otherwise false.
  def upload_file(file_path)
    @object.upload_file(file_path)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
#{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Laden Sie eine Datei mithilfe von `Object.put` hoch.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
    File.open(source_file_path, "rb") do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why:
#{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Laden Sie eine Datei mithilfe von `Object.put` hoch und fügen Sie eine serverseitige Verschlüsselung hinzu.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{@object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
    object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
    #{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK for Ruby API-Referenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
pub async fn upload_object(
    client: &aws_sdk_s3::Client,
    bucket_name: &str,
    file_name: &str,
    key: &str,
) -> Result<aws_sdk_s3::operation::put_object::PutObjectOutput, S3ExampleError> {
    let body =
aws_sdk_s3::primitives::ByteStream::from_path(std::path::Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await
        .map_err(S3ExampleError::from)
}
```

- API Einzelheiten finden Sie [PutObject](#) unter AWS SDK als API Referenz auf Rust.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
"Get contents of file from application server."  
DATA lv_body TYPE xstring.  
OPEN DATASET iv_file_name FOR INPUT IN BINARY MODE.  
READ DATASET iv_file_name INTO lv_body.  
CLOSE DATASET iv_file_name.  
  
"Upload/put an object to an S3 bucket."  
TRY.  
  lo_s3->putobject(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_file_name  
    iv_body = lv_body  
  ).  
  MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK als SAP ABAP API Referenz.

Swift

SDK für Swift

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import AWSS3  
import Smithy  
  
public func uploadFile(bucket: String, key: String, file: String) async  
throws {  
  let fileUrl = URL(fileURLWithPath: file)  
  do {  
    let fileData = try Data(contentsOf: fileUrl)
```

```
        let dataStream = ByteStream.data(fileData)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )

        _ = try await client.putObject(input: input)
    }
    catch {
        print("ERROR: ", dump(error, name: "Putting an object. "))
        throw error
    }
}
```

```
import AWSS3
import Smithy

public func createFile(bucket: String, key: String, withData data: Data)
async throws {
    let dataStream = ByteStream.data(data)

    let input = PutObjectInput(
        body: dataStream,
        bucket: bucket,
        key: key
    )

    do {
        _ = try await client.putObject(input: input)
    }
    catch {
        print("ERROR: ", dump(error, name: "Putting an object. "))
        throw error
    }
}
```

- API-Einheiten finden Sie [PutObject](#) unter AWS SDK als API-Swift-Referenz.

Für eine vollständige Liste von AWS SDKEntwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **PutObjectAcl** mit einem AWS SDKoder CLI

Die folgenden Codebeispiele zeigen, wie man es benutztPutObjectAcl.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Zugriffskontrolllisten verwalten \(\) ACLs](#)

C++

SDKfür C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
bool AwsDoc::S3::putObjectAcl(const Aws::String &bucketName, const Aws::String
    &objectKey, const Aws::String &ownerID,
                                const Aws::String &granteePermission, const
    Aws::String &granteeType,
                                const Aws::String &granteeID, const Aws::String
    &granteeEmailAddress,
                                const Aws::String &granteeURI, const
    Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::Owner owner;
    owner.SetID(ownerID);

    Aws::S3::Model::Grantee grantee;
    grantee.SetType(setGranteeType(granteeType));

    if (!granteeEmailAddress.empty()) {
        grantee.SetEmailAddress(granteeEmailAddress);
    }
}
```

```
    }

    if (!granteeID.empty()) {
        grantee.SetID(granteeID);
    }

    if (!granteeURI.empty()) {
        grantee.SetURI(granteeURI);
    }

    Aws::S3::Model::Grant grant;
    grant.SetGrantee(grantee);
    grant.SetPermission(setGranteePermission(granteePermission));

    Aws::Vector<Aws::S3::Model::Grant> grants;
    grants.push_back(grant);

    Aws::S3::Model::AccessControlPolicy acp;
    acp.SetOwner(owner);
    acp.SetGrants(grants);

    Aws::S3::Model::PutObjectAclRequest request;
    request.SetAccessControlPolicy(acp);
    request.SetBucket(bucketName);
    request.SetKey(objectKey);

    Aws::S3::Model::PutObjectAclOutcome outcome =
        s3Client.PutObjectAcl(request);

    if (!outcome.IsSuccess()) {
        auto error = outcome.GetError();
        std::cerr << "Error: putObjectAcl: " << error.GetExceptionName()
            << " - " << error.GetMessage() << std::endl;
    } else {
        std::cout << "Successfully added an ACL to the object '" << objectKey
            << "' in the bucket '" << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

/*! Routine which converts a human-readable string to a built-in type
    enumeration.
    /*!
```

```

    \param access: Human readable string.
    \return Permission: Permission enumeration.
    */
    Aws::S3::Model::Permission setGranteePermission(const Aws::String &access) {
        if (access == "FULL_CONTROL")
            return Aws::S3::Model::Permission::FULL_CONTROL;
        if (access == "WRITE")
            return Aws::S3::Model::Permission::WRITE;
        if (access == "READ")
            return Aws::S3::Model::Permission::READ;
        if (access == "WRITE_ACP")
            return Aws::S3::Model::Permission::WRITE_ACP;
        if (access == "READ_ACP")
            return Aws::S3::Model::Permission::READ_ACP;
        return Aws::S3::Model::Permission::NOT_SET;
    }

    /*! Routine which converts a human-readable string to a built-in type
    enumeration.
    */
    \param type: Human readable string.
    \return Type: Type enumeration.
    */
    Aws::S3::Model::Type setGranteeType(const Aws::String &type) {
        if (type == "Amazon customer by email")
            return Aws::S3::Model::Type::AmazonCustomerByEmail;
        if (type == "Canonical user")
            return Aws::S3::Model::Type::CanonicalUser;
        if (type == "Group")
            return Aws::S3::Model::Type::Group;
        return Aws::S3::Model::Type::NOT_SET;
    }
}

```

- API-Einheiten finden Sie [PutObjectAcl](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl gewährt `full control` zwei AWS Benutzer (`user1@example.com` und `user2@example.com`) und die `read` Erlaubnis für alle:

```
aws s3api put-object-acl --bucket MyBucket --key file.txt --grant-full-control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Einzelheiten zu Benutzerdefiniert finden Sie unter <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> ACLs (die ACL s3api-Befehle verwenden z. put-object-acl B. dieselbe Kurzschreibweise für Argumente).

- Einzelheiten finden Sie unter API [PutObjectAcl](#) AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
        in Boto3
                               that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key

    def put_acl(self, email):
        """
        Applies an ACL to the object that grants read access to an AWS user
        identified
        by email address.

        :param email: The email address of the user to grant access.
        """
```

```
try:
    acl = self.object.Acl()
    # Putting an ACL overwrites the existing ACL, so append new grants
    # if you want to preserve existing grants.
    grants = acl.grants if acl.grants else []
    grants.append(
        {
            "Grantee": {"Type": "AmazonCustomerByEmail", "EmailAddress":
email},
            "Permission": "READ",
        }
    )
    acl.put(AccessControlPolicy={"Grants": grants, "Owner": acl.owner})
    logger.info("Granted read access to %s.", email)
except ClientError:
    logger.exception("Couldn't add ACL to object '%s'.", self.object.key)
    raise
```

- API-Einheiten finden Sie [PutObjectAcl](#) unter AWS SDK für Python (Boto3) API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutObjectLegalHold** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutObjectLegalHold`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Amazon S3 S3-Objekte sperren](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Set or modify a legal hold on an object in an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The key of the object.</param>
/// <param name="holdStatus">The On or Off status for the legal hold.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectLegalHold(string bucketName,
    string objectKey, ObjectLockLegalHoldStatus holdStatus)
{
    try
    {
        var request = new PutObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            LegalHold = new ObjectLockLegalHold()
            {
                Status = holdStatus
            }
        };

        var response = await _amazonS3.PutObjectLegalHoldAsync(request);
        Console.WriteLine($"\\tModified legal hold for {objectKey} in
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tError modifying legal hold: '{ex.Message}'");
        return false;
    }
}
```

```
}
```

- API-Einheiten finden Sie [PutObjectLegalHold](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um einem Objekt eine gesetzliche Sperre zuzuweisen

Im folgenden `put-object-legal-hold` Beispiel wird ein Legal Hold für das Objekt `festgelegtdoc1.rtf`.

```
aws s3api put-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --legal-hold Status=ON
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutObjectLegalHold](#) unter AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.  
type S3Actions struct {  
  S3Client *s3.Client  
  S3Manager *manager.Uploader  
}
```

```
// PutObjectLegalHold sets the legal hold configuration for an S3 object.
func (actor S3Actions) PutObjectLegalHold(ctx context.Context, bucket string, key
string, versionId string, legalHoldStatus types.ObjectLockLegalHoldStatus) error
{
input := &s3.PutObjectLegalHoldInput{
    Bucket: aws.String(bucket),
    Key:    aws.String(key),
    LegalHold: &types.ObjectLockLegalHold{
        Status: legalHoldStatus,
    },
}
if versionId != "" {
    input.VersionId = aws.String(versionId)
}

_, err := actor.S3Client.PutObjectLegalHold(ctx, input)
if err != nil {
    var noKey *types.NoSuchKey
    if errors.As(err, &noKey) {
        log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
        err = noKey
    }
}

return err
}
```

- API-Einheiten finden Sie [PutObjectLegalHold](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// Set or modify a legal hold on an object in an S3 bucket.
public void modifyObjectLegalHold(String bucketName, String objectKey,
boolean legalHoldOn) {
    ObjectLockLegalHold legalHold ;
    if (legalHoldOn) {
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.ON)
            .build();
    } else {
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.OFF)
            .build();
    }

    PutObjectLegalHoldRequest legalHoldRequest =
PutObjectLegalHoldRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .legalHold(legalHold)
        .build();

    getClient().putObjectLegalHold(legalHoldRequest) ;
    System.out.println("Modified legal hold for "+ objectKey +" in
"+bucketName +".");
}
```

- API-Einheiten finden Sie [PutObjectLegalHold](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import { fileURLToPath } from "url";
import { PutObjectLegalHoldCommand, S3Client } from "@aws-sdk/client-s3";
```

```
/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
  const command = new PutObjectLegalHoldCommand({
    Bucket: bucketName,
    Key: objectKey,
    LegalHold: {
      // Set the status to 'ON' to place a legal hold on the object.
      // Set the status to 'OFF' to remove the legal hold.
      Status: "ON",
    },
    // Optionally, you can provide additional parameters
    // ChecksumAlgorithm: "ALGORITHM",
    // ContentMD5: "MD5_HASH",
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    // VersionId: "OBJECT_VERSION_ID",
  });

  try {
    const response = await client.send(command);
    console.log(
      `Object legal hold status: ${response.$metadata.httpStatusCode}`,
    );
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME", "OBJECT_KEY");
}
```

- API-Einheiten finden Sie [PutObjectLegalHold](#) unter AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Verschaffen Sie einem Objekt eine gesetzliche Sperre.

```
def set_legal_hold(s3_client, bucket: str, key: str) -> None:
    """
    Set a legal hold on a specific file in a bucket.

    Args:
        s3_client: Boto3 S3 client.
        bucket: The name of the bucket containing the file.
        key: The key of the file to set the legal hold on.
    """
    print()
    logger.info("Setting legal hold on file [%s] in bucket [%s]", key, bucket)
    try:
        before_status = "OFF"
        after_status = "ON"
        s3_client.put_object_legal_hold(
            Bucket=bucket, Key=key, LegalHold={"Status": after_status}
        )
        logger.debug(
            "Legal hold set successfully on file [%s] in bucket [%s]", key,
            bucket
        )
        _print_legal_hold_update(bucket, key, before_status, after_status)
    except Exception as e:
        logger.error(
            "Failed to set legal hold on file [%s] in bucket [%s]: %s", key,
            bucket, e
        )
```

- API-Einheiten finden Sie [PutObjectLegalHold](#) in AWS SDK für Python (Boto3) API - Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **PutObjectLockConfiguration** mit einem AWS SDK oder CLI


Die folgenden Codebeispiele zeigen, wie man es benutzt `PutObjectLockConfiguration`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Amazon S3 S3-Objekte sperren](#)

.NET

AWS SDK for .NET

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Legt die Objektsperrekonfiguration eines Buckets fest.

```
/// <summary>
/// Enable object lock on an existing bucket.
/// </summary>
/// <param name="bucketName">The name of the bucket to modify.</param>
/// <returns>True if successful.</returns>
public async Task<bool> EnableObjectLockOnBucket(string bucketName)
{
    try
    {
        // First, enable Versioning on the bucket.
        await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
```

```

    {
        BucketName = bucketName,
        VersioningConfig = new S3BucketVersioningConfig()
        {
            EnableMfaDelete = false,
            Status = VersionStatus.Enabled
        }
    });

    var request = new PutObjectLockConfigurationRequest()
    {
        BucketName = bucketName,
        ObjectLockConfiguration = new ObjectLockConfiguration()
        {
            ObjectLockEnabled = new ObjectLockEnabled("Enabled"),
        },
    };

    var response = await
    _amazonS3.PutObjectLockConfigurationAsync(request);
    Console.WriteLine($"{bucketName}\tAdded an object lock policy to bucket
{bucketName}.");
    return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}
catch (AmazonS3Exception ex)
{
    Console.WriteLine($"Error modifying object lock: '{ex.Message}'");
    return false;
}
}

```

Legen Sie die Standardaufbewahrungsdauer eines Buckets fest.

```

/// <summary>
/// Set or modify a retention period on an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket to modify.</param>
/// <param name="retention">The retention mode.</param>
/// <param name="retainUntilDate">The date for retention until.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyBucketDefaultRetention(string bucketName, bool
enableObjectLock, ObjectLockRetentionMode retention, DateTime retainUntilDate)

```



```
{
    var enabledString = enableObjectLock ? "Enabled" : "Disabled";
    var timeDifference = retainUntilDate.Subtract(DateTime.Now);
    try
    {
        // First, enable Versioning on the bucket.
        await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
        {
            BucketName = bucketName,
            VersioningConfig = new S3BucketVersioningConfig()
            {
                EnableMfaDelete = false,
                Status = VersionStatus.Enabled
            }
        });

        var request = new PutObjectLockConfigurationRequest()
        {
            BucketName = bucketName,
            ObjectLockConfiguration = new ObjectLockConfiguration()
            {
                ObjectLockEnabled = new ObjectLockEnabled(enabledString),
                Rule = new ObjectLockRule()
                {
                    DefaultRetention = new DefaultRetention()
                    {
                        Mode = retention,
                        Days = timeDifference.Days // Can be specified in
days or years but not both.
                    }
                }
            }
        };

        var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
        Console.WriteLine($"{\tAdded a default retention to bucket
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"{\tError modifying object lock: '{ex.Message}'");
    }
}
```

```
        return false;
    }
}
```

- API-Einheiten finden Sie [PutObjectLockConfiguration](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um eine Objektsperrenkonfiguration für einen Bucket festzulegen

Im folgenden `put-object-lock-configuration` Beispiel wird eine 50-tägige Objektsperre für den angegebenen Bucket eingerichtet.

```
aws s3api put-object-lock-configuration \
  --bucket my-bucket-with-object-lock \
  --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
  { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [PutObjectLockConfiguration](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Legt die Objektsperrenkonfiguration eines Buckets fest.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
```

```
S3Client *s3.Client
S3Manager *manager.Uploader
}

// EnableObjectLockOnBucket enables object locking on an existing bucket.
func (actor S3Actions) EnableObjectLockOnBucket(ctx context.Context, bucket
string) error {
// Versioning must be enabled on the bucket before object locking is enabled.
verInput := &s3.PutBucketVersioningInput{
Bucket: aws.String(bucket),
VersioningConfiguration: &types.VersioningConfiguration{
MFADelete: types.MFADeleteDisabled,
Status: types.BucketVersioningStatusEnabled,
},
}
_, err := actor.S3Client.PutBucketVersioning(ctx, verInput)
if err != nil {
var noBucket *types.NoSuchBucket
if errors.As(err, &noBucket) {
log.Printf("Bucket %s does not exist.\n", bucket)
err = noBucket
}
return err
}

input := &s3.PutObjectLockConfigurationInput{
Bucket: aws.String(bucket),
ObjectLockConfiguration: &types.ObjectLockConfiguration{
ObjectLockEnabled: types.ObjectLockEnabledEnabled,
},
}
_, err = actor.S3Client.PutObjectLockConfiguration(ctx, input)
if err != nil {
var noBucket *types.NoSuchBucket
if errors.As(err, &noBucket) {
log.Printf("Bucket %s does not exist.\n", bucket)
err = noBucket
}
}

return err
}
```

Legen Sie die Standardaufbewahrungsdauer eines Buckets fest.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
    S3Client    *s3.Client
    S3Manager   *manager.Uploader
}

// ModifyDefaultBucketRetention modifies the default retention period of an
// existing bucket.
func (actor S3Actions) ModifyDefaultBucketRetention(
    ctx context.Context, bucket string, lockMode types.ObjectLockEnabled,
    retentionPeriod int32, retentionMode types.ObjectLockRetentionMode) error {

    input := &s3.PutObjectLockConfigurationInput{
        Bucket: aws.String(bucket),
        ObjectLockConfiguration: &types.ObjectLockConfiguration{
            ObjectLockEnabled: lockMode,
            Rule: &types.ObjectLockRule{
                DefaultRetention: &types.DefaultRetention{
                    Days: aws.Int32(retentionPeriod),
                    Mode: retentionMode,
                },
            },
        },
    }

    _, err := actor.S3Client.PutObjectLockConfiguration(ctx, input)
    if err != nil {
        var noBucket *types.NoSuchBucket
        if errors.As(err, &noBucket) {
            log.Printf("Bucket %s does not exist.\n", bucket)
            err = noBucket
        }
    }

    return err
}
```

- API-Einheiten finden Sie [PutObjectLockConfiguration](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Legt die Objektsperrenkonfiguration eines Buckets fest.

```
// Enable object lock on an existing bucket.
public void enableObjectLockOnBucket(String bucketName) {
    try {
        VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
            .status(BucketVersioningStatus.ENABLED)
            .build();

        PutBucketVersioningRequest putBucketVersioningRequest =
PutBucketVersioningRequest.builder()
            .bucket(bucketName)
            .versioningConfiguration(versioningConfiguration)
            .build();

        // Enable versioning on the bucket.
        getClient().putBucketVersioning(putBucketVersioningRequest);
        PutObjectLockConfigurationRequest request =
PutObjectLockConfigurationRequest.builder()
            .bucket(bucketName)
            .objectLockConfiguration(ObjectLockConfiguration.builder()
                .objectLockEnabled(ObjectLockEnabled.ENABLED)
                .build())
            .build();

        getClient().putObjectLockConfiguration(request);
    }
}
```

```
        System.out.println("Successfully enabled object lock on
"+bucketName);

        } catch (S3Exception ex) {
            System.out.println("Error modifying object lock: '" + ex.getMessage()
+ "'");
        }
    }
}
```

Legen Sie die Standardaufbewahrungsdauer eines Buckets fest.

```
// Set or modify a retention period on an S3 bucket.
public void modifyBucketDefaultRetention(String bucketName) {
    VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
        .mfaDelete(MFADelete.DISABLED)
        .status(BucketVersioningStatus.ENABLED)
        .build();

    PutBucketVersioningRequest versioningRequest =
PutBucketVersioningRequest.builder()
        .bucket(bucketName)
        .versioningConfiguration(versioningConfiguration)
        .build();

    getClient().putBucketVersioning(versioningRequest);
    DefaultRetention rention = DefaultRetention.builder()
        .days(1)
        .mode(ObjectLockRetentionMode.GOVERNANCE)
        .build();

    ObjectLockRule lockRule = ObjectLockRule.builder()
        .defaultRetention(rention)
        .build();

    ObjectLockConfiguration objectLockConfiguration =
ObjectLockConfiguration.builder()
        .objectLockEnabled(ObjectLockEnabled.ENABLED)
        .rule(lockRule)
        .build();
}
```

```
PutObjectLockConfigurationRequest putObjectLockConfigurationRequest =
PutObjectLockConfigurationRequest.builder()
    .bucket(bucketName)
    .objectLockConfiguration(objectLockConfiguration)
    .build();

getClient().putObjectLockConfiguration(putObjectLockConfigurationRequest) ;
System.out.println("Added a default retention to bucket "+bucketName
+ ".");
}
```

- API-Einheiten finden Sie [PutObjectLockConfiguration](#) unter AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Legt die Objektsperrekonfiguration eines Buckets fest.

```
import { fileURLToPath } from "url";
import {
  PutObjectLockConfigurationCommand,
  S3Client,
} from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 */
export const main = async (client, bucketName) => {
  const command = new PutObjectLockConfigurationCommand({
    Bucket: bucketName,
```

```

    // The Object Lock configuration that you want to apply to the specified
    bucket.
    ObjectLockConfiguration: {
      ObjectLockEnabled: "Enabled",
    },
    // Optionally, you can provide additional parameters
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    // Token: "OPTIONAL_TOKEN",
  });

  try {
    const response = await client.send(command);
    console.log(
      `Object Lock Configuration updated: ${response.$metadata.httpStatusCode}`,
    );
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME");
}

```

Legen Sie die Standardaufbewahrungsdauer eines Buckets fest.

```

import { fileURLToPath } from "url";
import {
  PutObjectLockConfigurationCommand,
  S3Client,
} from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 */
export const main = async (client, bucketName) => {
  const command = new PutObjectLockConfigurationCommand({
    Bucket: bucketName,

```



```
// The Object Lock configuration that you want to apply to the specified
bucket.
ObjectLockConfiguration: {
  ObjectLockEnabled: "Enabled",
  Rule: {
    DefaultRetention: {
      Mode: "GOVERNANCE",
      Years: 3,
    },
  },
},
// Optionally, you can provide additional parameters
// ExpectedBucketOwner: "ACCOUNT_ID",
// RequestPayer: "requester",
// Token: "OPTIONAL_TOKEN",
});

try {
  const response = await client.send(command);
  console.log(
    `Default Object Lock Configuration updated: ${response.
$metadata.httpStatusCode}`,
  );
} catch (err) {
  console.error(err);
}
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME");
}
```

- API-Einheiten finden Sie [PutObjectLockConfiguration](#) unter AWS SDK for JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Setzen Sie die Objektsperren-Konfiguration.

```
s3_client.put_object_lock_configuration(  
    Bucket=bucket,  
    ObjectLockConfiguration={"ObjectLockEnabled": "Disabled", "Rule":  
    {}},  
)
```

- API Einzelheiten finden Sie [PutObjectLockConfiguration](#) in AWS SDK für Python (Boto3) API - Referenz.

Für eine vollständige Liste von AWS SDK Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Verwenden Sie **PutObjectRetention** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `PutObjectRetention`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Amazon S3 S3-Objekte sperren](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/// <summary>
/// Set or modify a retention period on an object in an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The key of the object.</param>
/// <param name="retention">The retention mode.</param>
/// <param name="retainUntilDate">The date retention expires.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectRetentionPeriod(string bucketName,
    string objectKey, ObjectLockRetentionMode retention, DateTime
retainUntilDate)
{
    try
    {
        var request = new PutObjectRetentionRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            Retention = new ObjectLockRetention()
            {
                Mode = retention,
                RetainUntilDate = retainUntilDate
            }
        };

        var response = await _amazonS3.PutObjectRetentionAsync(request);
        Console.WriteLine($"{objectKey} in {bucketName}
until {retainUntilDate:d}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
```

```
        Console.WriteLine($"\\tError modifying retention period:
'{ex.Message}');
        return false;
    }
}
```

- API-Einheiten finden Sie [PutObjectRetention](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um eine Objektaufbewahrungskonfiguration für ein Objekt festzulegen

Im folgenden `put-object-retention` Beispiel wird eine Objektaufbewahrungskonfiguration für das angegebene Objekt bis zum 01.01.2025 festgelegt.

```
aws s3api put-object-retention \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf \
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate":
  "2025-01-01T00:00:00" }'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [PutObjectRetention](#) AWS CLI Befehlsreferenz.

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// S3Actions wraps S3 service actions.
```

```
type S3Actions struct {
    S3Client    *s3.Client
    S3Manager   *manager.Uploader
}

// PutObjectRetention sets the object retention configuration for an S3 object.
func (actor S3Actions) PutObjectRetention(ctx context.Context, bucket string, key
string, retentionMode types.ObjectLockRetentionMode, retentionPeriodDays int32)
error {
    input := &s3.PutObjectRetentionInput{
        Bucket: aws.String(bucket),
        Key:    aws.String(key),
        Retention: &types.ObjectLockRetention{
            Mode:          retentionMode,
            RetainUntilDate: aws.Time(time.Now().AddDate(0, 0, int(retentionPeriodDays))),
        },
        BypassGovernanceRetention: aws.Bool(true),
    }

    _, err := actor.S3Client.PutObjectRetention(ctx, input)
    if err != nil {
        var noKey *types.NoSuchKey
        if errors.As(err, &noKey) {
            log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
            err = noKey
        }
    }

    return err
}
```

- API-Einheiten finden Sie [PutObjectRetention](#) unter AWS SDK for Go API-Referenz.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
// Set or modify a retention period on an object in an S3 bucket.
public void modifyObjectRetentionPeriod(String bucketName, String objectKey)
{
    // Calculate the instant one day from now.
    Instant futureInstant = Instant.now().plus(1, ChronoUnit.DAYS);

    // Convert the Instant to a ZonedDateTime object with a specific time
    zone.
    ZonedDateTime zonedDateTime =
futureInstant.atZone(ZoneId.systemDefault());

    // Define a formatter for human-readable output.
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss");

    // Format the ZonedDateTime object to a human-readable date string.
    String humanReadableDate = formatter.format(zonedDateTime);

    // Print the formatted date string.
    System.out.println("Formatted Date: " + humanReadableDate);
    ObjectLockRetention retention = ObjectLockRetention.builder()
        .mode(ObjectLockRetentionMode.GOVERNANCE)
        .retainUntilDate(futureInstant)
        .build();

    PutObjectRetentionRequest retentionRequest =
PutObjectRetentionRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .retention(retention)
        .build();
}
```

```
getClient().putObjectRetention(retentionRequest);
System.out.println("Set retention for "+objectKey +" in " +bucketName +"
until "+ humanReadableDate +".");
}
```

- API Einzelheiten finden Sie [PutObjectRetention](#) unter AWS SDK for Java 2.x API Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import { fileURLToPath } from "url";
import { PutObjectRetentionCommand, S3Client } from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
  const command = new PutObjectRetentionCommand({
    Bucket: bucketName,
    Key: objectKey,
    BypassGovernanceRetention: false,
    // ChecksumAlgorithm: "ALGORITHM",
    // ContentMD5: "MD5_HASH",
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    Retention: {
      Mode: "GOVERNANCE", // or "COMPLIANCE"
      RetainUntilDate: new Date(new Date().getTime() + 24 * 60 * 60 * 1000),
    },
    // VersionId: "OBJECT_VERSION_ID",
  });
```

```
try {
  const response = await client.send(command);
  console.log(
    `Object Retention settings updated: ${response.$metadata.httpStatusCode}`,
  );
} catch (err) {
  console.error(err);
}
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME", "OBJECT_KEY");
}
```

- API-Einheiten finden Sie [PutObjectRetention](#) unter AWS SDK for JavaScript API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Der Befehl aktiviert den Governance-Aufbewahrungsmodus bis zum Datum „31. Dezember 2019 00:00:00“ für das Objekt 'testfile.txt' im angegebenen S3-Bucket.

```
Write-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt' -
Retention_Mode GOVERNANCE -Retention_RetainUntilDate "2019-12-31T00:00:00"
```

- Einzelheiten finden Sie unter API [PutObjectRetention](#) AWS Tools for PowerShell Cmdlet-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Legen Sie eine Objektaufbewahrung fest.

```
s3_client.put_object_retention(  
    Bucket=bucket,  
    Key=key,  
    VersionId=version_id,  
    Retention={"Mode": "GOVERNANCE", "RetainUntilDate":  
far_future_date},  
    BypassGovernanceRetention=True,  
)
```

- API-Einheiten finden Sie [PutObjectRetention](#) in AWS SDK für Python (Boto3) API - Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **RestoreObject** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `RestoreObject`.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu [GitHub](#). Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;  
using System.Threading.Tasks;  
using Amazon;  
using Amazon.S3;  
using Amazon.S3.Model;  
  
/// <summary>  
/// This example shows how to restore an archived object in an Amazon
```

```
/// Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class RestoreArchivedObject
{
    public static void Main()
    {
        string bucketName = "amzn-s3-demo-bucket";
        string objectKey = "archived-object.txt";

        // Specify your bucket region (an example region is shown).
        RegionEndpoint bucketRegion = RegionEndpoint.USWest2;

        IAmazonS3 client = new AmazonS3Client(bucketRegion);
        RestoreObjectAsync(client, bucketName, objectKey).Wait();
    }

    /// <summary>
    /// This method restores an archived object from an Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// RestoreObjectAsync.</param>
    /// <param name="bucketName">A string representing the name of the
    /// bucket where the object was located before it was archived.</param>
    /// <param name="objectKey">A string representing the name of the
    /// archived object to restore.</param>
    public static async Task RestoreObjectAsync(IAmazonS3 client, string
bucketName, string objectKey)
    {
        try
        {
            var restoreRequest = new RestoreObjectRequest
            {
                BucketName = bucketName,
                Key = objectKey,
                Days = 2,
            };
            RestoreObjectResponse response = await
client.RestoreObjectAsync(restoreRequest);

            // Check the status of the restoration.
            await CheckRestorationStatusAsync(client, bucketName, objectKey);
        }
        catch (AmazonS3Exception amazonS3Exception)
    }
}
```

```

        {
            Console.WriteLine($"Error: {amazonS3Exception.Message}");
        }
    }

    /// <summary>
    /// This method retrieves the status of the object's restoration.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// GetObjectMetadataAsync.</param>
    /// <param name="bucketName">A string representing the name of the Amazon
    /// S3 bucket which contains the archived object.</param>
    /// <param name="objectKey">A string representing the name of the
    /// archived object you want to restore.</param>
    public static async Task CheckRestorationStatusAsync(IAmazonS3 client,
string bucketName, string objectKey)
    {
        GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
        };

        GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);

        var restStatus = response.RestoreInProgress ? "in-progress" :
"finished or failed";
        Console.WriteLine($"Restoration status: {restStatus}");
    }
}

```

- API-Einheiten finden Sie [RestoreObject](#) unter AWS SDK for .NET API-Referenz.

CLI

AWS CLI

Um eine Wiederherstellungsanforderung für ein Objekt zu erstellen

Im folgenden `restore-object` Beispiel wird das angegebene Amazon S3 Glacier-Objekt für den Bucket `my-glacier-bucket` für 10 Tage wiederhergestellt.

```
aws s3api restore-object \  
  --bucket my-glacier-bucket \  
  --key doc1.rtf \  
  --restore-request Days=10
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [RestoreObject](#) unter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.s3.S3Client;  
import software.amazon.awssdk.services.s3.model.RestoreRequest;  
import software.amazon.awssdk.services.s3.model.GlacierJobParameters;  
import software.amazon.awssdk.services.s3.model.RestoreObjectRequest;  
import software.amazon.awssdk.services.s3.model.S3Exception;  
import software.amazon.awssdk.services.s3.model.Tier;  
  
/*  
 * For more information about restoring an object, see "Restoring an archived  
 * object" at  
 * https://docs.aws.amazon.com/AmazonS3/latest/userguide/restoring-objects.html  
 *  
 * Before running this Java V2 code example, set up your development  
 * environment, including your credentials.  
 *  
 * For more information, see the following documentation topic:  
 *
```

```
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class RestoreObject {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <bucketName> <keyName> <expectedBucketOwner>

            Where:
                bucketName - The Amazon S3 bucket name.\s
                keyName - The key name of an object with a Storage class
value of Glacier.\s
                expectedBucketOwner - The account that owns the bucket (you
can obtain this value from the AWS Management Console).\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        String expectedBucketOwner = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        restoreS3Object(s3, bucketName, keyName, expectedBucketOwner);
        s3.close();
    }

    public static void restoreS3Object(S3Client s3, String bucketName, String
keyName, String expectedBucketOwner) {
        try {
            RestoreRequest restoreRequest = RestoreRequest.builder()
                .days(10)

                .glacierJobParameters(GlacierJobParameters.builder().tier(Tier.STANDARD).build())
                .build();
        }
    }
}
```

```

        RestoreObjectRequest objectRequest = RestoreObjectRequest.builder()
            .expectedBucketOwner(expectedBucketOwner)
            .bucket(bucketName)
            .key(keyName)
            .restoreRequest(restoreRequest)
            .build();

        s3.restoreObject(objectRequest);

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

- API-Einheiten finden Sie [RestoreObject](#) unter AWS SDK for Java 2.x API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **SelectObjectContent** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie man es benutzt `SelectObjectContent`.

CLI

AWS CLI

Um den Inhalt eines Amazon S3 S3-Objekts auf der Grundlage einer SQL-Anweisung zu filtern

Das folgende `select-object-content` Beispiel filtert das Objekt `my-data-file.csv` mit der angegebenen SQL-Anweisung und sendet die Ausgabe in eine Datei.

```

aws s3api select-object-content \
  --bucket my-bucket \
  --key my-data-file.csv \
  --expression "select * from s3object limit 100" \
  --expression-type 'SQL' \
  --input-serialization '{"CSV": {}, "CompressionType": "NONE"}' \

```

```
--output-serialization '{"CSV": {}}' "output.csv"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SelectObjectContent](#) unter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Das folgende Beispiel zeigt eine Abfrage, die ein JSON Objekt verwendet. Das [vollständige Beispiel](#) zeigt auch die Verwendung eines CSV Objekts.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.model.CSVInput;
import software.amazon.awssdk.services.s3.model.CSVOutput;
import software.amazon.awssdk.services.s3.model.CompressionType;
import software.amazon.awssdk.services.s3.model.ExpressionType;
import software.amazon.awssdk.services.s3.model.FileHeaderInfo;
import software.amazon.awssdk.services.s3.model.InputSerialization;
import software.amazon.awssdk.services.s3.model.JSONInput;
import software.amazon.awssdk.services.s3.model.JSONOutput;
import software.amazon.awssdk.services.s3.model.JSONType;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.services.s3.model.OutputSerialization;
import software.amazon.awssdk.services.s3.model.Progress;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.SelectObjectContentRequest;
import
    software.amazon.awssdk.services.s3.model.SelectObjectContentResponseHandler;
```

```
import software.amazon.awssdk.services.s3.model.Stats;

import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

public class SelectObjectContentExample {
    static final Logger logger =
        LoggerFactory.getLogger(SelectObjectContentExample.class);
    static final String BUCKET_NAME = "select-object-content-" +
        UUID.randomUUID();
    static final S3AsyncClient s3AsyncClient = S3AsyncClient.create();
    static String FILE_CSV = "csv";
    static String FILE_JSON = "json";
    static String URL_CSV = "https://raw.githubusercontent.com/mledoze/countries/
master/dist/countries.csv";
    static String URL_JSON = "https://raw.githubusercontent.com/mledoze/
countries/master/dist/countries.json";

    public static void main(String[] args) {
        SelectObjectContentExample selectObjectContentExample = new
        SelectObjectContentExample();
        try {
            SelectObjectContentExample.setUp();
            selectObjectContentExample.runSelectObjectContentMethodForJSON();
            selectObjectContentExample.runSelectObjectContentMethodForCSV();
        } catch (SdkException e) {
            logger.error(e.getMessage(), e);
            System.exit(1);
        } finally {
            SelectObjectContentExample.tearDown();
        }
    }

    EventStreamInfo runSelectObjectContentMethodForJSON() {
        // Set up request parameters.
        final String queryExpression = "select * from s3object[*][*] c where
c.area < 350000";
        final String fileType = FILE_JSON;

        InputSerialization inputSerialization = InputSerialization.builder()
```



```
        .json(JSONInput.builder().type(JSOType.DOCUMENT).build())
        .compressionType(CompressionType.NONE)
        .build();

    OutputSerialization outputSerialization = OutputSerialization.builder()
        .json(JSONOutput.builder().recordDelimiter(null).build())
        .build();

    // Build the SelectObjectContentRequest.
    SelectObjectContentRequest select = SelectObjectContentRequest.builder()
        .bucket(BUCKET_NAME)
        .key(FILE_JSON)
        .expression(queryExpression)
        .expressionType(ExpressionType.SQL)
        .inputSerialization(inputSerialization)
        .outputSerialization(outputSerialization)
        .build();

    EventStreamInfo eventStreamInfo = new EventStreamInfo();
    // Call the selectObjectContent method with the request and a response
    handler.
    // Supply an EventStreamInfo object to the response handler to gather
    records and information from the response.
    s3AsyncClient.selectObjectContent(select,
    buildResponseHandler(eventStreamInfo)).join();

    // Log out information gathered while processing the response stream.
    long recordCount = eventStreamInfo.getRecords().stream().mapToInt(record
    ->
        record.split("\n").length
    ).sum();
    logger.info("Total records {}: {}", fileType, recordCount);
    logger.info("Visitor onRecords for fileType {} called {} times",
    fileType, eventStreamInfo.getCountOnRecordsCalled());
    logger.info("Visitor onStats for fileType {}, {}", fileType,
    eventStreamInfo.getStats());
    logger.info("Visitor onContinuations for fileType {}, {}", fileType,
    eventStreamInfo.getCountContinuationEvents());
    return eventStreamInfo;
}

    static SelectObjectContentResponseHandler
    buildResponseHandler(EventStreamInfo eventStreamInfo) {
```

```

    // Use a Visitor to process the response stream. This visitor logs
    information and gathers details while processing.
    final SelectObjectContentResponseHandler.Visitor visitor =
    SelectObjectContentResponseHandler.Visitor.builder()
        .onRecords(r -> {
            logger.info("Record event received.");
            eventStreamInfo.addRecord(r.payload().asUtf8String());
            eventStreamInfo.incrementOnRecordsCalled();
        })
        .onCont(ce -> {
            logger.info("Continuation event received.");
            eventStreamInfo.incrementContinuationEvents();
        })
        .onProgress(pe -> {
            Progress progress = pe.details();
            logger.info("Progress event received:\n bytesScanned:
{} \n bytesProcessed: {} \n bytesReturned: {}",
                progress.bytesScanned(),
                progress.bytesProcessed(),
                progress.bytesReturned());
        })
        .onEnd(ee -> logger.info("End event received."))
        .onStats(se -> {
            logger.info("Stats event received.");
            eventStreamInfo.addStats(se.details());
        })
        .build();

    // Build the SelectObjectContentResponseHandler with the visitor that
    processes the stream.
    return SelectObjectContentResponseHandler.builder()
        .subscriber(visitor).build();
}

// The EventStreamInfo class is used to store information gathered while
processing the response stream.
static class EventStreamInfo {
    private final List<String> records = new ArrayList<>();
    private Integer countOnRecordsCalled = 0;
    private Integer countContinuationEvents = 0;
    private Stats stats;

    void incrementOnRecordsCalled() {
        countOnRecordsCalled++;
    }
}

```

```
    }

    void incrementContinuationEvents() {
        countContinuationEvents++;
    }

    void addRecord(String record) {
        records.add(record);
    }

    void addStats(Stats stats) {
        this.stats = stats;
    }

    public List<String> getRecords() {
        return records;
    }

    public Integer getCountOnRecordsCalled() {
        return countOnRecordsCalled;
    }

    public Integer getCountContinuationEvents() {
        return countContinuationEvents;
    }

    public Stats getStats() {
        return stats;
    }
}
```

- API-Einheiten finden Sie [SelectObjectContent](#) in AWS SDK for Java 2.x API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **UploadPart** mit einem AWS SDK oder CLI


Die folgenden Codebeispiele zeigen, wie man es benutzt `UploadPart`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Durchführen eines mehrteiligen Uploads](#)
- [Verwenden der Prüfsummen](#)
- [Arbeiten Sie mit Amazon S3 S3-Objektintegrität](#)

C++

SDK für C++

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```

//! Upload a part to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param partNumber:
    \param checksumAlgorithm: Checksum algorithm, ignored when NOT_SET.
    \param calculatedHash: A data integrity hash to set, depending on the
checksum algorithm,
                           ignored when it is an empty string.
    \param body: An shared_ptr IOStream of the data to be uploaded.
    \param client: The S3 client instance used to perform the upload operation.
    \return UploadPartOutcome: The outcome.
*/

Aws::S3::Model::UploadPartOutcome AwsDoc::S3::uploadPart(const Aws::String
&bucket,
                                                         const Aws::String &key,
                                                         const Aws::String
&uploadID,
                                                         int partNumber,
                                                         Aws::S3::Model::ChecksumAlgorithm checksumAlgorithm,

```

```

const Aws::String
&calculatedHash,
const
std::shared_ptr<Aws::IOStream> &body,
const Aws::S3::S3Client
&client) {
    Aws::S3::Model::UploadPartRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);
    request.SetPartNumber(partNumber);
    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }
    request.SetBody(body);

    if (!calculatedHash.empty()) {
        switch (checksumAlgorithm) {
            case Aws::S3::Model::ChecksumAlgorithm::NOT_SET:
                request.SetContentMD5(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32:
                request.SetChecksumCRC32(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32C:
                request.SetChecksumCRC32C(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA1:
                request.SetChecksumSHA1(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA256:
                request.SetChecksumSHA256(calculatedHash);
                break;
        }
    }

    return client.UploadPart(request);
}

```

- API-Einheiten finden Sie [UploadPart](#) unter AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Der folgende Befehl lädt den ersten Teil eines mehrteiligen Uploads hoch, der `create-multipart-upload` mit dem Befehl initiiert wurde:

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --  
body part01 --upload-id  
"dFRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P"
```

Die `body` Option verwendet den Namen oder Pfad einer lokalen Datei für den Upload (verwenden Sie nicht das Präfix `file://`). Die Mindestteilgröße beträgt 5 MB. Die Upload-ID wird von zurückgegeben `create-multipart-upload` und kann auch mit abgerufen werden `list-multipart-uploads`. Bucket und Schlüssel werden angegeben, wenn Sie den mehrteiligen Upload erstellen.

Ausgabe:

```
{  
  "ETag": "\"e868e0f4719e394144ef36531ee6824c\""  
}
```

Speichern Sie den ETag Wert jedes Teils für später. Sie sind erforderlich, um den mehrteiligen Upload abzuschließen.

- API-Einheiten finden Sie unter [UploadPart](#) AWS CLI Befehlsreferenz.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
let mut upload_parts: Vec<aws_sdk_s3::types::CompletedPart> = Vec::new();
```

```
for chunk_index in 0..chunk_count {
  let this_chunk = if chunk_count - 1 == chunk_index {
    size_of_last_chunk
  } else {
    CHUNK_SIZE
  };
  let stream = ByteStream::read_from()
    .path(path)
    .offset(chunk_index * CHUNK_SIZE)
    .length(Length::Exact(this_chunk))
    .build()
    .await
    .unwrap();

  // Chunk index needs to start at 0, but part numbers start at 1.
  let part_number = (chunk_index as i32) + 1;
  let upload_part_res = client
    .upload_part()
    .key(&key)
    .bucket(&bucket_name)
    .upload_id(upload_id)
    .body(stream)
    .part_number(part_number)
    .send()
    .await?;

  upload_parts.push(
    CompletedPart::builder()
      .e_tag(upload_part_res.e_tag.unwrap_or_default())
      .part_number(part_number)
      .build(),
  );
}
```

```
// Create a multipart upload. Use UploadPart and CompleteMultipartUpload to
// upload the file.
let multipart_upload_res: CreateMultipartUploadOutput = client
  .create_multipart_upload()
  .bucket(&bucket_name)
  .key(&key)
  .send()
  .await?;
```

```
let upload_id = multipart_upload_res.upload_id().ok_or(S3ExampleError::new(
    "Missing upload_id after CreateMultipartUpload",
))?;
```

```
// upload_parts: Vec<aws_sdk_s3::types::CompletedPart>
let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
    .set_parts(Some(upload_parts))
    .build();

let _complete_multipart_upload_res = client
    .complete_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .multipart_upload(completed_multipart_upload)
    .upload_id(upload_id)
    .send()
    .await?;
```

- API-Einheiten finden Sie [UploadPart](#) unter AWS SDK als API Referenz auf Rust.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für die Verwendung von Amazon S3 AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie gängige Szenarien in Amazon S3 implementieren können mit AWS SDKs. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben erledigen können, indem Sie mehrere Funktionen innerhalb von Amazon S3 oder in Kombination mit anderen aufrufen AWS-Services. Jedes Szenario enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zur Einrichtung und Ausführung des Codes finden.

Die Szenarien zielen auf ein mittleres Erfahrungsniveau ab, damit Sie Serviceaktionen im Kontext besser verstehen können.

Beispiele

- [Konvertieren Sie Text in Sprache und zurück in Text mit einem AWS SDK](#)
- [Erstellen Sie eine URL für Amazon S3 vorsignierte Datei mit einem AWS SDK](#)
- [Eine Anwendung für Foto-Asset-Management erstellen, mit der Benutzer Fotos mithilfe von Labels verwalten können](#)
- [Eine Webseite, die Amazon S3 S3-Objekte mithilfe eines auflistet AWS SDK](#)
- [Erstellen Sie eine Amazon-Text-Explorer-Anwendung](#)
- [Löschen Sie alle Objekte in einem bestimmten Amazon S3 S3-Bucket mit einem AWS SDK.](#)
- [Löschen Sie unvollständige mehrteilige Uploads auf Amazon S3 mit einem AWS SDK](#)
- [Erkennung PPE in Bildern mit Amazon Rekognition mithilfe eines AWS SDK](#)
- [Erkennen Sie Entitäten in Text, der aus einem Bild extrahiert wurde, mithilfe eines AWS SDK](#)
- [Erkennen Sie Gesichter in einem Bild mit einem AWS SDK](#)
- [Erkennen Sie Objekte in Bildern mit Amazon Rekognition mithilfe eines AWS SDK](#)
- [Erkennen Sie Personen und Objekte in einem Video mit Amazon Rekognition mithilfe eines AWS SDK](#)
- [Herunterladen aller Objekte aus einem Amazon Simple Storage Service \(Amazon S3\)-Bucket in ein lokales Verzeichnis](#)
- [Rufen Sie ein Amazon S3 S3-Objekt von einem Access Point mit mehreren Regionen ab, indem Sie eine AWS SDK](#)
- [Holen Sie sich ein Objekt aus einem Amazon S3 S3-Bucket mit einem AWS SDK, unter Angabe eines If-Modified-Since-Headers](#)
- [Erste Schritte mit der Verschlüsselung von Amazon S3 S3-Objekten mithilfe eines AWS SDK](#)
- [Erste Schritte mit Tags für Amazon S3 S3-Objekte mithilfe eines AWS SDK](#)
- [Arbeiten Sie mit den Objektsperrefunktionen von Amazon S3 mithilfe eines AWS SDK](#)
- [Verwalten Sie Zugriffskontrolllisten \(ACLs\) für Amazon S3 S3-Buckets mit einem AWS SDK](#)
- [Verwalten Sie versionierte Amazon S3 S3-Objekte stapelweise mit einer Lambda-Funktion mithilfe eines AWS SDK](#)
- [Analysieren Sie Amazon S3 URIs mit einem AWS SDK](#)
- [Erstellen Sie eine mehrteilige Kopie eines Amazon S3 S3-Objekts mit einem AWS SDK](#)
- [Führen Sie einen mehrteiligen Upload eines Amazon S3 S3-Objekts mit einem AWS SDK](#)
- [Empfangen und verarbeiten Sie Amazon S3 S3-Ereignisbenachrichtigungen mithilfe eines AWS SDK.](#)

- [Speichern EXIF und andere Bildinformationen mithilfe eines AWS SDK](#)
- [Senden Sie S3-Ereignisbenachrichtigungen an Amazon EventBridge mit einem AWS SDK](#)
- [Verfolgen Sie den Upload oder Download eines Amazon S3 S3-Objekts mithilfe eines AWS SDK](#)
- [Transformieren Sie Daten für Ihre Anwendung mit S3 Object Lambda](#)
- [Beispielansätze für Einheiten- und Integrationstests mit einem AWS SDK](#)
- [Rekursives Hochladen eines lokalen Verzeichnisses in einen Amazon Simple Storage Service \(Amazon S3\)-Bucket](#)
- [Laden Sie große Dateien zu und von Amazon S3 hoch oder laden Sie sie herunter mit einem AWS SDK](#)
- [Laden Sie einen Stream unbekannter Größe mit einem in ein Amazon S3 S3-Objekt hoch AWS SDK](#)
- [Verwenden Sie Prüfsummen, um mit einem Amazon S3 S3-Objekt zu arbeiten, indem Sie AWS SDK](#)
- [Arbeiten Sie mit den Objektintegritätsfunktionen von Amazon S3 mithilfe eines AWS SDK](#)
- [Arbeiten Sie mit Objekten mit Amazon S3 S3-Version unter Verwendung eines AWS SDK](#)

Konvertieren Sie Text in Sprache und zurück in Text mit einem AWS SDK

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Verwenden Sie Amazon Polly, um eine Nur-Text-Eingabedatei (UTF-8) zu einer Audiodatei zu synthetisieren.
- Laden Sie die Audiodatei in einen Amazon-S3-Bucket hoch.
- Konvertieren Sie die Audiodatei mit Amazon Transcribe in Text.
- Zeigen Sie den Text an.

Rust

SDK für Rust

Verwenden Sie Amazon Polly, um eine Nur-Text-Eingabedatei (UTF-8) in eine Audiodatei zu synthetisieren, die Audiodatei in einen Amazon S3 S3-Bucket hochzuladen, Amazon Transcribe zu verwenden, um diese Audiodatei in Text zu konvertieren und den Text anzuzeigen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter. [GitHub](#)

In diesem Beispiel verwendete Dienste

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erstellen Sie eine URL für Amazon S3 vorgezeichnete Datei mit einem AWS SDK

Die folgenden Codebeispiele zeigen, wie Sie eine URL für Amazon S3 vorgezeichnetes Objekt erstellen und ein Objekt hochladen.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu [GitHub](#). Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Generieren Sie ein vorgezeichnetes URL, das für eine begrenzte Zeit eine Amazon S3 S3-Aktion ausführen kann.

```
using System;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

public class GenPresignedUrl
{
    public static void Main()
```

```
{
    const string bucketName = "amzn-s3-demo-bucket";
    const string objectKey = "sample.txt";

    // Specify how long the presigned URL lasts, in hours
    const double timeoutDuration = 12;

    // Specify the AWS Region of your Amazon S3 bucket. If it is
    // different from the Region defined for the default user,
    // pass the Region to the constructor for the client. For
    // example: new AmazonS3Client(RegionEndpoint.USEast1);

    // If using the Region us-east-1, and server-side encryption with AWS
    KMS, you must specify Signature Version 4.
    // Region us-east-1 defaults to Signature Version 2 unless explicitly
    set to Version 4 as shown below.
    // For more details, see https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingAWSSDK.html#specify-signature-version
    // and https://docs.aws.amazon.com/sdkfornet/v3/apidocs/items/Amazon/TAWSConfigsS3.html
    AWSConfigsS3.UseSignatureVersion4 = true;
    IAmazonS3 s3Client = new AmazonS3Client(RegionEndpoint.USEast1);

    string urlString = GeneratePresignedURL(s3Client, bucketName,
objectKey, timeoutDuration);
    Console.WriteLine($"The generated URL is: {urlString}.");
}

/// <summary>
/// Generate a presigned URL that can be used to access the file named
/// in the objectKey parameter for the amount of time specified in the
/// duration parameter.
/// </summary>
/// <param name="client">An initialized S3 client object used to call
/// the GetPresignedUrl method.</param>
/// <param name="bucketName">The name of the S3 bucket containing the
/// object for which to create the presigned URL.</param>
/// <param name="objectKey">The name of the object to access with the
/// presigned URL.</param>
/// <param name="duration">The length of time for which the presigned
/// URL will be valid.</param>
/// <returns>A string representing the generated presigned URL.</returns>
public static string GeneratePresignedURL(IAmazonS3 client, string
bucketName, string objectKey, double duration)
```

```
{
    string urlString = string.Empty;
    try
    {
        var request = new GetPreSignedUrlRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            Expires = DateTime.UtcNow.AddHours(duration),
        };
        urlString = client.GetPreSignedURL(request);
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error: '{ex.Message}'");
    }

    return urlString;
}
}
```

Generieren Sie ein vorsigniertes URL und führen Sie damit einen Upload durch. URL

```
using System;
using System.IO;
using System.Net.Http;
using System.Threading.Tasks;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to upload an object to an Amazon Simple Storage
/// Service (Amazon S3) bucket using a presigned URL. The code first
/// creates a presigned URL and then uses it to upload an object to an
/// Amazon S3 bucket using that URL.
/// </summary>
public class UploadUsingPresignedURL
{
    private static HttpClient httpClient = new HttpClient();
```

```
public static async Task Main()
{
    string bucketName = "amzn-s3-demo-bucket";
    string keyName = "samplefile.txt";
    string filePath = $"source\\{keyName}";

    // Specify how long the signed URL will be valid in hours.
    double timeoutDuration = 12;

    // Specify the AWS Region of your Amazon S3 bucket. If it is
    // different from the Region defined for the default user,
    // pass the Region to the constructor for the client. For
    // example: new AmazonS3Client(RegionEndpoint.USEast1);

    // If using the Region us-east-1, and server-side encryption with AWS
    KMS, you must specify Signature Version 4.
    // Region us-east-1 defaults to Signature Version 2 unless explicitly
    set to Version 4 as shown below.
    // For more details, see https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingAWSSDK.html#specify-signature-version
    // and https://docs.aws.amazon.com/sdkfornet/v3/apidocs/items/Amazon/TAWSConfigsS3.html
    AWSConfigsS3.UseSignatureVersion4 = true;
    IAmazonS3 client = new AmazonS3Client(RegionEndpoint.USEast1);

    var url = GeneratePreSignedURL(client, bucketName, keyName,
    timeoutDuration);
    var success = await UploadObject(filePath, url);

    if (success)
    {
        Console.WriteLine("Upload succeeded.");
    }
    else
    {
        Console.WriteLine("Upload failed.");
    }
}

/// <summary>
/// Uploads an object to an Amazon S3 bucket using the presigned URL
passed in
/// the url parameter.
/// </summary>
```

```
/// <param name="filePath">The path (including file name) to the local
/// file you want to upload.</param>
/// <param name="url">The presigned URL that will be used to upload the
/// file to the Amazon S3 bucket.</param>
/// <returns>A Boolean value indicating the success or failure of the
/// operation, based on the HttpResponseMessage.</returns>
public static async Task<bool> UploadObject(string filePath, string url)
{
    using var streamContent = new StreamContent(
        new FileStream(filePath, FileMode.Open, FileAccess.Read));

    var response = await httpClient.PutAsync(url, streamContent);
    return response.IsSuccessStatusCode;
}

/// <summary>
/// Generates a presigned URL which will be used to upload an object to
/// an Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used to
call
/// GetPreSignedURL.</param>
/// <param name="bucketName">The name of the Amazon S3 bucket to which
the
/// presigned URL will point.</param>
/// <param name="objectKey">The name of the file that will be uploaded.</
param>
/// <param name="duration">How long (in hours) the presigned URL will
/// be valid.</param>
/// <returns>The generated URL.</returns>
public static string GeneratePreSignedURL(
    IAmazonS3 client,
    string bucketName,
    string objectKey,
    double duration)
{
    var request = new GetPreSignedUrlRequest
    {
        BucketName = bucketName,
        Key = objectKey,
        Verb = HttpVerb.PUT,
        Expires = DateTime.UtcNow.AddHours(duration),
    };
};
```

```
        string url = client.GetPreSignedURL(request);
        return url;
    }
}
```

C++

SDK für C++

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Generieren Sie ein vorsigniertes ObjektURL, um ein Objekt herunterzuladen.

```
//! Routine which demonstrates creating a pre-signed URL to download an object
from an
//! Amazon Simple Storage Service (Amazon S3) bucket.
/*!
    \param bucketName: Name of the bucket.
    \param key: Name of an object key.
    \param expirationSeconds: Expiration in seconds for pre-signed URL.
    \param clientConfig: Aws client configuration.
    \return Aws::String: A pre-signed URL.
*/
Aws::String AwsDoc::S3::generatePreSignedGetObjectUrl(const Aws::String
&bucketName,
                                                    const Aws::String &key,
                                                    uint64_t expirationSeconds,
                                                    const
Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    return client.GeneratePresignedUrl(bucketName, key,
    Aws::Http::HttpMethod::HTTP_GET,
                                                    expirationSeconds);
}
```


Mit libcurl herunterladen.

```
static size_t myCurlWriteBack(char *buffer, size_t size, size_t nitems, void
*userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    if (nitems > 0) {
        str->write(buffer, size * nitems);
    }
    return size * nitems;
}

//! Utility routine to test getObject with a pre-signed URL.
/*!
 \param presignedURL: A pre-signed URL to get an object from a bucket.
 \param resultString: A string to hold the result.
 \return bool: Function succeeded.
*/
bool AwsDoc::S3::getObjectWithPresignedObjectUrl(const Aws::String &presignedURL,
                                                  Aws::String &resultString) {

    CURL *curl = curl_easy_init();
    CURLcode result;

    std::stringstream outWriteString;

    result = curl_easy_setopt(curl, CURLOPT_WRITEDATA, &outWriteString);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEDATA " << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, myCurlWriteBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEFUNCTION" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_URL, presignedURL.c_str());

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_URL" << std::endl;
        return false;
    }
}
```

```

    }

    result = curl_easy_perform(curl);

    if (result != CURLE_OK) {
        std::cerr << "Failed to perform CURL request" << std::endl;
        return false;
    }

    resultString = outWriteString.str();

    if (resultString.find("<?xml") == 0) {
        std::cerr << "Failed to get object, response:\n" << resultString <<
std::endl;
        return false;
    }

    return true;
}

```

Generieren Sie ein vorsigniertes ObjektURL, um ein Objekt hochzuladen.

```

/*! Routine which demonstrates creating a pre-signed URL to upload an object to
an
Amazon Simple Storage Service (Amazon S3) bucket.
*/
\param bucketName: Name of the bucket.
\param key: Name of an object key.
\param clientConfig: Aws client configuration.
\return Aws::String: A pre-signed URL.
*/
Aws::String AwsDoc::S3::generatePreSignedPutObjectUrl(const Aws::String
&bucketName,
                                                    const Aws::String &key,
                                                    uint64_t expirationSeconds,
                                                    const
Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    return client.GeneratePresignedUrl(bucketName, key,
    Aws::Http::HttpMethod::HTTP_PUT,
                                                    expirationSeconds);
}

```

Mit libcurl hochladen.

```
static size_t myCurlReadBack(char *buffer, size_t size, size_t nitems, void
*userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    str->read(buffer, size * nitems);

    return str->gcount();
}

static size_t myCurlWriteBack(char *buffer, size_t size, size_t nitems, void
*userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    if (nitems > 0) {
        str->write(buffer, size * nitems);
    }
    return size * nitems;
}

//! Utility routine to test putObject with a pre-signed URL.
/*!
    \param presignedURL: A pre-signed URL to put an object in a bucket.
    \param data: Body of the putObject request.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::PutStringWithPresignedObjectURL(const Aws::String &presignedURL,
                                                  const Aws::String &data) {

    CURL *curl = curl_easy_init();
    CURLcode result;

    Aws::StringStream readStringStream;
    readStringStream << data;
    result = curl_easy_setopt(curl, CURLOPT_READFUNCTION, myCurlReadBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_READFUNCTION" << std::endl;
        return false;
    }
}
```

```
result = curl_easy_setopt(curl, CURLOPT_READDATA, &readStringStream);
if (result != CURLE_OK) {
    std::cerr << "Failed to set CURLOPT_READDATA" << std::endl;
    return false;
}

result = curl_easy_setopt(curl, CURLOPT_INFILESIZE_LARGE,
                          (curl_off_t) data.size());

if (result != CURLE_OK) {
    std::cerr << "Failed to set CURLOPT_INFILESIZE_LARGE" << std::endl;
    return false;
}

result = curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, myCurlWriteBack);

if (result != CURLE_OK) {
    std::cerr << "Failed to set CURLOPT_WRITEFUNCTION" << std::endl;
    return false;
}

std::stringstream outWriteString;

result = curl_easy_setopt(curl, CURLOPT_WRITEDATA, &outWriteString);

if (result != CURLE_OK) {
    std::cerr << "Failed to set CURLOPT_WRITEDATA " << std::endl;
    return false;
}

result = curl_easy_setopt(curl, CURLOPT_URL, presignedURL.c_str());

if (result != CURLE_OK) {
    std::cerr << "Failed to set CURLOPT_URL" << std::endl;
    return false;
}

result = curl_easy_setopt(curl, CURLOPT_UPLOAD, 1L);

if (result != CURLE_OK) {
    std::cerr << "Failed to set CURLOPT_PUT" << std::endl;
    return false;
}
```

```
    result = curl_easy_perform(curl);

    if (result != CURLE_OK) {
        std::cerr << "Failed to perform CURL request" << std::endl;
        return false;
    }

    std::string outString = outWriteString.str();
    if (outString.empty()) {
        std::cout << "Successfully put object." << std::endl;
        return true;
    } else {
        std::cout << "A server error was encountered, output:\n" << outString
            << std::endl;
        return false;
    }
}
```

Go

SDK für Go V2

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie Funktionen, die S3-Vorsignierungsaktionen umschließen.

```
// Presigner encapsulates the Amazon Simple Storage Service (Amazon S3) presign
actions
// used in the examples.
// It contains PresignClient, a client that is used to presign requests to Amazon
S3.
// Presigned requests contain temporary credentials and can be made from any HTTP
client.
type Presigner struct {
    PresignClient *s3.PresignClient
}
```

```
// GetObject makes a presigned request that can be used to get an object from a
// bucket.
// The presigned request is valid for the specified number of seconds.
func (presigner Presigner) GetObject(
    bucketName string, objectKey string, lifetimeSecs int64)
(*v4.PresignedHTTPRequest, error) {
    request, err := presigner.PresignClient.PresignGetObject(context.TODO(),
    &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    }, func(opts *s3.PresignOptions) {
        opts.Expires = time.Duration(lifetimeSecs * int64(time.Second))
    })
    if err != nil {
        log.Printf("Couldn't get a presigned request to get %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return request, err
}

// PutObject makes a presigned request that can be used to put an object in a
// bucket.
// The presigned request is valid for the specified number of seconds.
func (presigner Presigner) PutObject(
    bucketName string, objectKey string, lifetimeSecs int64)
(*v4.PresignedHTTPRequest, error) {
    request, err := presigner.PresignClient.PresignPutObject(context.TODO(),
    &s3.PutObjectInput{
        Bucket: aws.String(bucketName),
        Key:    aws.String(objectKey),
    }, func(opts *s3.PresignOptions) {
        opts.Expires = time.Duration(lifetimeSecs * int64(time.Second))
    })
    if err != nil {
        log.Printf("Couldn't get a presigned request to put %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return request, err
}
```

```
// DeleteObject makes a presigned request that can be used to delete an object
// from a bucket.
func (presigner Presigner) DeleteObject(bucketName string, objectKey string)
(*v4.PresignedHTTPRequest, error) {
    request, err := presigner.PresignClient.PresignDeleteObject(context.TODO(),
&s3.DeleteObjectInput{
    Bucket: aws.String(bucketName),
    Key:    aws.String(objectKey),
})
    if err != nil {
        log.Printf("Couldn't get a presigned request to delete object %v. Here's why:
%v\n", objectKey, err)
    }
    return request, err
}
```

Führen Sie ein interaktives Beispiel aus, das Presigned generiert und verwendet, URLs um ein S3-Objekt hochzuladen, herunterzuladen und zu löschen.

```
// RunPresigningScenario is an interactive example that shows you how to get
// presigned
// HTTP requests that you can use to move data into and out of Amazon Simple
// Storage
// Service (Amazon S3). The presigned requests contain temporary credentials and
// can
// be used by an HTTP client.
//
// 1. Get a presigned request to put an object in a bucket.
// 2. Use the net/http package to use the presigned request to upload a local
// file to the bucket.
// 3. Get a presigned request to get an object from a bucket.
// 4. Use the net/http package to use the presigned request to download the
// object to a local file.
// 5. Get a presigned request to delete an object from a bucket.
// 6. Use the net/http package to use the presigned request to delete the object.
//
```

```
// This example creates an Amazon S3 presign client from the specified sdkConfig
// so that
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
// example.
// This package can be found in the ..\..\demotools folder of this repo.
//
// It uses an IHttpRequester interface to abstract HTTP requests so they can be
// mocked
// during testing.
func RunPresigningScenario(sdkConfig aws.Config, questioner
demotools.IQuestioner, httpRequester IHttpRequester) {
defer func() {
if r := recover(); r != nil {
fmt.Printf("Something went wrong with the demo.")
}
}()

log.Println(strings.Repeat("-", 88))
log.Println("Welcome to the Amazon S3 presigning demo.")
log.Println(strings.Repeat("-", 88))

s3Client := s3.NewFromConfig(sdkConfig)
bucketBasics := actions.BucketBasics{S3Client: s3Client}
presignClient := s3.NewPresignClient(s3Client)
presigner := actions.Presigner{PresignClient: presignClient}

bucketName := questioner.Ask("We'll need a bucket. Enter a name for a bucket "+
"you own or one you want to create:", demotools.NotEmpty{})
bucketExists, err := bucketBasics.BucketExists(bucketName)
if err != nil {
panic(err)
}
if !bucketExists {
err = bucketBasics.CreateBucket(bucketName, sdkConfig.Region)
if err != nil {
panic(err)
} else {
log.Println("Bucket created.")
}
}
log.Println(strings.Repeat("-", 88))
```



```
log.Printf("Let's presign a request to upload a file to your bucket.")
uploadFilename := questioner.Ask("Enter the path to a file you want to upload:",
    demotools.NotEmpty{})
uploadKey := questioner.Ask("What would you like to name the uploaded object?",
    demotools.NotEmpty{})
uploadFile, err := os.Open(uploadFilename)
if err != nil {
    panic(err)
}
defer uploadFile.Close()
presignedPutRequest, err := presigner.PutObject(bucketName, uploadKey, 60)
if err != nil {
    panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
    presignedPutRequest.Method,
    presignedPutRequest.URL)
log.Println("Using net/http to send the request...")
info, err := uploadFile.Stat()
if err != nil {
    panic(err)
}
putResponse, err := httpRequester.Put(presignedPutRequest.URL, info.Size(),
    uploadFile)
if err != nil {
    panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.",
    presignedPutRequest.Method,
    uploadKey, putResponse.StatusCode)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's presign a request to download the object.")
questioner.Ask("Press Enter when you're ready.")
presignedGetRequest, err := presigner.GetObject(bucketName, uploadKey, 60)
if err != nil {
    panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
    presignedGetRequest.Method,
    presignedGetRequest.URL)
log.Println("Using net/http to send the request...")
getResponse, err := httpRequester.Get(presignedGetRequest.URL)
if err != nil {
```

```
panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.",
presignedGetRequest.Method,
uploadKey, getResponse.StatusCode)
defer getResponse.Body.Close()
downloadBody, err := io.ReadAll(getResponse.Body)
if err != nil {
panic(err)
}
log.Printf("Downloaded %v bytes. Here are the first 100 of them:\n",
len(downloadBody))
log.Println(strings.Repeat("-", 88))
log.Println(string(downloadBody[:100]))
log.Println(strings.Repeat("-", 88))

log.Println("Let's presign a request to delete the object.")
questioner.Ask("Press Enter when you're ready.")
presignedDelRequest, err := presigner.DeleteObject(bucketName, uploadKey)
if err != nil {
panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
presignedDelRequest.Method,
presignedDelRequest.URL)
log.Println("Using net/http to send the request...")
delResponse, err := httpRequester.Delete(presignedDelRequest.URL)
if err != nil {
panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.\n",
presignedDelRequest.Method,
uploadKey, delResponse.StatusCode)
log.Println(strings.Repeat("-", 88))

log.Println("Thanks for watching!")
log.Println(strings.Repeat("-", 88))
}
```

Definieren Sie einen HTTP Anfrage-Wrapper, der im Beispiel verwendet wird, um Anfragen zu stellenHTTP.

```
// IHttpRequester abstracts HTTP requests into an interface so it can be mocked
// during
// unit testing.
type IHttpRequester interface {
    Get(url string) (resp *http.Response, err error)
    Put(url string, contentLength int64, body io.Reader) (resp *http.Response, err
    error)
    Delete(url string) (resp *http.Response, err error)
}

// HttpRequester uses the net/http package to make HTTP requests during the
// scenario.
type HttpRequester struct{}

func (httpReq HttpRequester) Get(url string) (resp *http.Response, err error) {
    return http.Get(url)
}
func (httpReq HttpRequester) Put(url string, contentLength int64, body io.Reader)
(resp *http.Response, err error) {
    putRequest, err := http.NewRequest("PUT", url, body)
    if err != nil {
        return nil, err
    }
    putRequest.ContentLength = contentLength
    return http.DefaultClient.Do(putRequest)
}
func (httpReq HttpRequester) Delete(url string) (resp *http.Response, err error)
{
    delRequest, err := http.NewRequest("DELETE", url, nil)
    if err != nil {
        return nil, err
    }
    return http.DefaultClient.Do(delRequest)
}
```

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Generieren Sie eine URL für ein Objekt vorsignierte Datei und laden Sie sie dann herunter (GETAnfrage).

Importiert.

```
import com.example.s3.util.PresignUrlUtils;
import org.slf4j.Logger;
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import
    software.amazon.awssdk.services.s3.presigner.model.GetObjectPresignRequest;
import
    software.amazon.awssdk.services.s3.presigner.model.PresignedGetObjectRequest;
import software.amazon.awssdk.utils.IoUtils;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URISyntaxException;
import java.net.URL;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
```

```
import java.nio.file.Paths;
import java.time.Duration;
import java.util.UUID;
```

Generieren Sie die URL.

```
/* Create a pre-signed URL to download an object in a subsequent GET request.
*/
public String createPresignedGetUrl(String bucketName, String keyName) {
    try (S3Presigner presigner = S3Presigner.create()) {

        GetObjectRequest objectRequest = GetObjectRequest.builder()
            .bucket(bucketName)
            .key(keyName)
            .build();

        GetObjectPresignRequest presignRequest =
        GetObjectPresignRequest.builder()
            .signatureDuration(Duration.ofMinutes(10)) // The URL will
expire in 10 minutes.
            .getObjectRequest(objectRequest)
            .build();

        PresignedGetObjectRequest presignedRequest =
presigner.presignGetObject(presignRequest);
        logger.info("Presigned URL: [{}]",
presignedRequest.url().toString());
        logger.info("HTTP method: [{}]",
presignedRequest.httpRequest().method());

        return presignedRequest.url().toExternalForm();
    }
}
```

Laden Sie das Objekt herunter, indem Sie einen der folgenden drei Ansätze verwenden.

Verwenden Sie die Klasse `JDK HttpURLConnection` (seit Version 1.1), um den Download durchzuführen.

```
/* Use the JDK HttpURLConnection (since v1.1) class to do the download. */
public byte[] useHttpURLConnectionToGet(String presignedUrlString) {
```

```
    ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(); // Capture the response body to a byte array.

    try {
        URL presignedUrl = new URL(presignedUrlString);
        HttpURLConnection connection = (HttpURLConnection)
presignedUrl.openConnection();
        connection.setRequestMethod("GET");
        // Download the result of executing the request.
        try (InputStream content = connection.getInputStream()) {
            IoUtils.copy(content, byteArrayOutputStream);
        }
        logger.info("HTTP response code is " + connection.getResponseCode());
    } catch (S3Exception | IOException e) {
        logger.error(e.getMessage(), e);
    }
    return byteArrayOutputStream.toByteArray();
}
```

Verwenden Sie die Klasse JDK `HttpClient` (seit v11), um den Download durchzuführen.

```
/* Use the JDK HttpClient (since v11) class to do the download. */
public byte[] useHttpClientToGet(String presignedUrlString) {
    ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(); // Capture the response body to a byte array.

    HttpRequest.Builder requestBuilder = HttpRequest.newBuilder();
    HttpClient httpClient = HttpClient.newHttpClient();
    try {
        URL presignedUrl = new URL(presignedUrlString);
        HttpResponse<InputStream> response = httpClient.send(requestBuilder
            .uri(presignedUrl.toURI())
            .GET()
            .build(),
            HttpResponse.BodyHandlers.ofInputStream());

        IoUtils.copy(response.body(), byteArrayOutputStream);

        logger.info("HTTP response code is " + response.statusCode());
    } catch (URISyntaxException | InterruptedException | IOException e) {
```

```

        logger.error(e.getMessage(), e);
    }
    return byteArrayOutputStream.toByteArray();
}

```

Benutze die AWS SDK für die `SdkHttpClient` Java-Klasse, um den Download durchzuführen.

```

/* Use the AWS SDK for Java SdkHttpClient class to do the download. */
public byte[] useSdkHttpClientToPut(String presignedUrlString) {

    ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(); // Capture the response body to a byte array.
    try {
        URL presignedUrl = new URL(presignedUrlString);
        SdkHttpRequest request = SdkHttpRequest.builder()
            .method(SdkHttpMethod.GET)
            .uri(presignedUrl.toURI())
            .build();

        HttpExecuteRequest executeRequest = HttpExecuteRequest.builder()
            .request(request)
            .build();

        try (SdkHttpClient sdkHttpClient = ApacheHttpClient.create()) {
            HttpExecuteResponse response =
sdkHttpClient.prepareRequest(executeRequest).call();
            response.responseBody().ifPresentOrElse(
                abortableInputStream -> {
                    try {
                        IoUtils.copy(abortableInputStream,
byteArrayOutputStream);
                    } catch (IOException e) {
                        throw new RuntimeException(e);
                    }
                },
                () -> logger.error("No response body."));

            logger.info("HTTP Response code is {}",
response.httpResponse().statusCode());
        }
    } catch (URISyntaxException | IOException e) {

```

```
        logger.error(e.getMessage(), e);
    }
    return byteArrayOutputStream.toByteArray();
}
```

Generieren Sie eine URL für einen Upload vorsignierte Datei und laden Sie dann eine Datei hoch (PUTAnfrage).

Importiert.

```
import com.example.s3.util.PresignUrlUtils;
import org.slf4j.Logger;
import software.amazon.awssdk.core.internal.sync.FileContentStreamProvider;
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import
    software.amazon.awssdk.services.s3.presigner.model.PresignedPutObjectRequest;
import
    software.amazon.awssdk.services.s3.presigner.model.PutObjectPresignRequest;

import java.io.File;
import java.io.IOException;
import java.io.OutputStream;
import java.io.RandomAccessFile;
import java.net.HttpURLConnection;
import java.net.URISyntaxException;
import java.net.URL;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import java.nio.file.Path;
import java.nio.file.Paths;
```



```
import java.time.Duration;
import java.util.Map;
import java.util.UUID;
```

Generieren Sie dieURL.

```
/* Create a presigned URL to use in a subsequent PUT request */
public String createPresignedUrl(String bucketName, String keyName,
Map<String, String> metadata) {
    try (S3Presigner presigner = S3Presigner.create()) {

        PutObjectRequest objectRequest = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(keyName)
            .metadata(metadata)
            .build();

        PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
            .signatureDuration(Duration.ofMinutes(10)) // The URL
expires in 10 minutes.
            .putObjectRequest(objectRequest)
            .build();

        PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);
        String myURL = presignedRequest.url().toString();
        logger.info("Presigned URL to upload a file to: [{}]", myURL);
        logger.info("HTTP method: [{}]",
presignedRequest.httpRequest().method());

        return presignedRequest.url().toExternalForm();
    }
}
```

Laden Sie ein Dateiojekt hoch, indem Sie einen der folgenden drei Ansätze verwenden.

Verwenden Sie die Klasse `JDK HttpURLConnection` (seit Version 1.1), um den Upload durchzuführen.

```

    /* Use the JDK HttpURLConnection (since v1.1) class to do the upload. */
    public void useHttpURLConnectionToPut(String presignedUrlString, File
fileToPut, Map<String, String> metadata) {
        logger.info("Begin [{}] upload", fileToPut.toString());
        try {
            URL presignedUrl = new URL(presignedUrlString);
            HttpURLConnection connection = (HttpURLConnection)
presignedUrl.openConnection();
            connection.setDoOutput(true);
            metadata.forEach((k, v) -> connection.setRequestProperty("x-amz-
meta-" + k, v));
            connection.setRequestMethod("PUT");
            OutputStream out = connection.getOutputStream();

            try (RandomAccessFile file = new RandomAccessFile(fileToPut, "r");
                FileChannel inChannel = file.getChannel()) {
                ByteBuffer buffer = ByteBuffer.allocate(8192); //Buffer size is
8k

                while (inChannel.read(buffer) > 0) {
                    buffer.flip();
                    for (int i = 0; i < buffer.limit(); i++) {
                        out.write(buffer.get());
                    }
                    buffer.clear();
                }
            } catch (IOException e) {
                logger.error(e.getMessage(), e);
            }

            out.close();
            connection.getResponseCode();
            logger.info("HTTP response code is " + connection.getResponseCode());

        } catch (S3Exception | IOException e) {
            logger.error(e.getMessage(), e);
        }
    }
}

```

Verwenden Sie die Klasse `JDK HttpClient` (seit v11), um den Upload durchzuführen.

```

/* Use the JDK HttpClient (since v11) class to do the upload. */

```

```

public void useHttpClientToPut(String presignedUrlString, File fileToPut,
Map<String, String> metadata) {
    logger.info("Begin [{}] upload", fileToPut.toString());

    HttpRequest.Builder requestBuilder = HttpRequest.newBuilder();
    metadata.forEach((k, v) -> requestBuilder.header("x-amz-meta-" + k, v));

    HttpClient httpClient = HttpClient.newHttpClient();
    try {
        final HttpResponse<Void> response = httpClient.send(requestBuilder
            .uri(new URL(presignedUrlString).toURI())
            .PUT(HttpRequest.BodyPublishers.ofFile(Path.of(fileToPut.toURI()))
                .build(),
                HttpResponse.BodyHandlers.discarding());

        logger.info("HTTP response code is " + response.statusCode());
    } catch (URISyntaxException | InterruptedException | IOException e) {
        logger.error(e.getMessage(), e);
    }
}

```

Benutze die AWS für die Java SdkHttpClient V2-Klasse, um den Upload durchzuführen.

```

/* Use the AWS SDK for Java V2 SdkHttpClient class to do the upload. */
public void useSdkHttpClientToPut(String presignedUrlString, File fileToPut,
Map<String, String> metadata) {
    logger.info("Begin [{}] upload", fileToPut.toString());

    try {
        URL presignedUrl = new URL(presignedUrlString);

        SdkHttpRequest.Builder requestBuilder = SdkHttpRequest.builder()
            .method(SdkHttpMethod.PUT)
            .uri(presignedUrl.toURI());
        // Add headers
        metadata.forEach((k, v) -> requestBuilder.putHeader("x-amz-meta-" +
k, v));
        // Finish building the request.
        SdkHttpRequest request = requestBuilder.build();
    }
}

```

```
HttpExecuteRequest executeRequest = HttpExecuteRequest.builder()
    .request(request)
    .contentStreamProvider(new
FileContentStreamProvider(fileToPut.toPath()))
    .build();

try (SdkHttpClient sdkHttpClient = ApacheHttpClient.create()) {
    HttpExecuteResponse response =
sdkHttpClient.prepareRequest(executeRequest).call();
    logger.info("Response code: {}",
response.httpResponse().statusCode());
}
} catch (URISyntaxException | IOException e) {
    logger.error(e.getMessage(), e);
}
}
```

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie ein vorsigniertes URL, um ein Objekt in einen Bucket hochzuladen.

```
import https from "https";
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { fromIni } from "@aws-sdk/credential-providers";
import { HttpRequest } from "@smithy/protocol-http";
import {
    getSignedUrl,
    S3RequestPresigner,
} from "@aws-sdk/s3-request-presigner";
import { parseUrl } from "@smithy/url-parser";
import { formatUrl } from "@aws-sdk/util-format-url";
import { Hash } from "@smithy/hash-node";
```

```
const createPresignedUrlWithoutClient = async ({ region, bucket, key }) => {
  const url = parseUrl(`https://${bucket}.s3.${region}.amazonaws.com/${key}`);
  const presigner = new S3RequestPresigner({
    credentials: fromIni(),
    region,
    sha256: Hash.bind(null, "sha256"),
  });

  const signedUrlObject = await presigner.presign(
    new HttpRequest({ ...url, method: "PUT" }),
  );
  return formatUrl(signedUrlObject);
};

const createPresignedUrlWithClient = ({ region, bucket, key }) => {
  const client = new S3Client({ region });
  const command = new PutObjectCommand({ Bucket: bucket, Key: key });
  return getSignedUrl(client, command, { expiresIn: 3600 });
};

function put(url, data) {
  return new Promise((resolve, reject) => {
    const req = https.request(
      url,
      { method: "PUT", headers: { "Content-Length": new Blob([data]).size } },
      (res) => {
        let responseBody = "";
        res.on("data", (chunk) => {
          responseBody += chunk;
        });
        res.on("end", () => {
          resolve(responseBody);
        });
      },
    );
    req.on("error", (err) => {
      reject(err);
    });
    req.write(data);
    req.end();
  });
}

export const main = async () => {
```

```
const REGION = "us-east-1";
const BUCKET = "example_bucket";
const KEY = "example_file.txt";

// There are two ways to generate a presigned URL.
// 1. Use createPresignedUrl without the S3 client.
// 2. Use getSignedUrl in conjunction with the S3 client and GetObjectCommand.
try {
  const noClientUrl = await createPresignedUrlWithoutClient({
    region: REGION,
    bucket: BUCKET,
    key: KEY,
  });

  const clientUrl = await createPresignedUrlWithClient({
    region: REGION,
    bucket: BUCKET,
    key: KEY,
  });

  // After you get the presigned URL, you can provide your own file
  // data. Refer to put() above.
  console.log("Calling PUT using presigned URL without client");
  await put(noClientUrl, "Hello World");

  console.log("Calling PUT using presigned URL with client");
  await put(clientUrl, "Hello World");

  console.log("\nDone. Check your S3 console.");
} catch (err) {
  console.error(err);
}
};
```

Erstellen Sie ein vorsigniertes ObjektURL, um ein Objekt aus einem Bucket herunterzuladen.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { fromIni } from "@aws-sdk/credential-providers";
import { HttpRequest } from "@smithy/protocol-http";
import {
  getSignedUrl,
  S3RequestPresigner,
}
```

```
} from "@aws-sdk/s3-request-presigner";
import { parseUrl } from "@smithy/url-parser";
import { formatUrl } from "@aws-sdk/util-format-url";
import { Hash } from "@smithy/hash-node";

const createPresignedUrlWithoutClient = async ({ region, bucket, key }) => {
  const url = parseUrl(`https://${bucket}.s3.${region}.amazonaws.com/${key}`);
  const presigner = new S3RequestPresigner({
    credentials: fromIni(),
    region,
    sha256: Hash.bind(null, "sha256"),
  });

  const signedUrlObject = await presigner.presign(new HttpRequest(url));
  return formatUrl(signedUrlObject);
};

const createPresignedUrlWithClient = ({ region, bucket, key }) => {
  const client = new S3Client({ region });
  const command = new GetObjectCommand({ Bucket: bucket, Key: key });
  return getSignedUrl(client, command, { expiresIn: 3600 });
};

export const main = async () => {
  const REGION = "us-east-1";
  const BUCKET = "example_bucket";
  const KEY = "example_file.jpg";

  try {
    const noClientUrl = await createPresignedUrlWithoutClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    const clientUrl = await createPresignedUrlWithClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    console.log("Presigned URL without client");
    console.log(noClientUrl);
    console.log("\n");
  }
}
```

```
    console.log("Presigned URL with client");
    console.log(clientUrl);
  } catch (err) {
    console.error(err);
  }
};
```

- Weitere Informationen finden Sie unter [AWS SDK for JavaScript Leitfaden](#) für Entwickler.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie eine `GetObject` vorsignierte Anfrage und verwenden Sie die URL, um ein Objekt herunterzuladen.

```
suspend fun getObjectPresigned(
    s3: S3Client,
    bucketName: String,
    keyName: String,
): String {
    // Create a GetObjectRequest.
    val unsignedRequest =
        GetObjectRequest {
            bucket = bucketName
            key = keyName
        }

    // Presign the GetObject request.
    val presignedRequest = s3.presignGetObject(unsignedRequest, 24.hours)

    // Use the URL from the presigned HttpRequest in a subsequent HTTP GET
    request to retrieve the object.
    val objectContents = URL(presignedRequest.url.toString()).readText()
```



```
    return objectContents
}
```

Erstellen Sie eine `GetObject` vorsignierte Anfrage mit erweiterten Optionen.

```
suspend fun getObjectPresignedMoreOptions(
    s3: S3Client,
    bucketName: String,
    keyName: String,
): HttpRequest {
    // Create a GetObjectRequest.
    val unsignedRequest =
        GetObjectRequest {
            bucket = bucketName
            key = keyName
        }

    // Presign the GetObject request.
    val presignedRequest =
        s3.presignGetObject(unsignedRequest, signer = CrtAwsSigner) {
            signingDate = Instant.now() + 12.hours // Presigned request can be
            used 12 hours from now.
            algorithm = AwsSigningAlgorithm.SIGV4_ASYMMETRIC
            signatureType = AwsSignatureType.HTTP_REQUEST_VIA_QUERY_PARAMS
            expiresAfter = 8.hours // Presigned request expires 8 hours later.
        }
    return presignedRequest
}
```

Erstellen Sie eine vorsignierte `PutObject`-Anfrage und verwenden Sie sie, um ein Objekt hochzuladen.

```
suspend fun putObjectPresigned(
    s3: S3Client,
    bucketName: String,
    keyName: String,
    content: String,
) {
    // Create a PutObjectRequest.
    val unsignedRequest =
```

```
PutObjectRequest {
    bucket = bucketName
    key = keyName
}

// Presign the request.
val presignedRequest = s3.presignPutObject(unsignedRequest, 24.hours)

// Use the URL and any headers from the presigned HttpRequest in a subsequent
HTTP PUT request to retrieve the object.
// Create a PUT request using the OkHttpClient API.
val putRequest =
    Request
        .Builder()
        .url(presignedRequest.url.toString())
        .apply {
            presignedRequest.headers.forEach { key, values ->
                header(key, values.joinToString(", "))
            }
        }.put(content.toRequestBody())
        .build()

val response = OkHttpClient().newCall(putRequest).execute()
assert(response.isSuccessful)
}
```

- Weitere Informationen finden Sie unter [AWS SDK für den Kotlin-Entwicklerhandbuch](#).

PHP

SDK für PHP

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
namespace S3;
use Aws\Exception\AwsException;
use AwsUtilities\PrintableLineBreak;
```

```
use AwsUtilities\TestableReadline;
use DateTime;

require 'vendor/autoload.php';

class PresignedURL
{
    use PrintableLineBreak;
    use TestableReadline;

    public function run()
    {
        $s3Service = new S3Service();

        $expiration = new DateTime("+20 minutes");
        $linebreak = $this->getLineBreak();

        echo $linebreak;
        echo ("Welcome to the Amazon S3 presigned URL demo.\n");
        echo $linebreak;

        $bucket = $this->testable_readline("First, please enter the name of the
S3 bucket to use: ");
        $key = $this->testable_readline("Next, provide the key of an object in
the given bucket: ");
        echo $linebreak;
        $command = $s3Service->getClient()->getCommand('GetObject', [
            'Bucket' => $bucket,
            'Key' => $key,
        ]);
        try {
            $preSignedUrl = $s3Service->preSignedUrl($command, $expiration);
            echo "Your preSignedUrl is \n$preSignedUrl\nand will be good for the
next 20 minutes.\n";
            echo $linebreak;
            echo "Thanks for trying the Amazon S3 presigned URL demo.\n";
        } catch (AwsException $exception) {
            echo $linebreak;
            echo "Something went wrong: $exception";
            die();
        }
    }
}
```

```
$runner = new PresignedURL();  
$runner->run();
```

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Generieren Sie ein vorsigniertes URL, das eine S3-Aktion für eine begrenzte Zeit ausführen kann. Verwenden Sie das Requests-Paket, um eine Anfrage mit dem URL zu stellen.

```
import argparse  
import logging  
import boto3  
from botocore.exceptions import ClientError  
import requests  
  
logger = logging.getLogger(__name__)  
  
def generate_presigned_url(s3_client, client_method, method_parameters,  
                           expires_in):  
    """  
    Generate a presigned Amazon S3 URL that can be used to perform an action.  
  
    :param s3_client: A Boto3 Amazon S3 client.  
    :param client_method: The name of the client method that the URL performs.  
    :param method_parameters: The parameters of the specified client method.  
    :param expires_in: The number of seconds the presigned URL is valid for.  
    :return: The presigned URL.  
    """  
    try:  
        url = s3_client.generate_presigned_url(  

```

```
        ClientMethod=client_method, Params=method_parameters,
ExpiresIn=expires_in
    )
    logger.info("Got presigned URL: %s", url)
except ClientError:
    logger.exception(
        "Couldn't get a presigned URL for client method '%s'.", client_method
    )
    raise
return url

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Welcome to the Amazon S3 presigned URL demo.")
    print("-" * 88)

    parser = argparse.ArgumentParser()
    parser.add_argument("bucket", help="The name of the bucket.")
    parser.add_argument(
        "key",
        help="For a GET operation, the key of the object in Amazon S3. For a "
        "PUT operation, the name of a file to upload.",
    )
    parser.add_argument("action", choices=("get", "put"), help="The action to
perform.")
    args = parser.parse_args()

    s3_client = boto3.client("s3")
    client_action = "get_object" if args.action == "get" else "put_object"
    url = generate_presigned_url(
        s3_client, client_action, {"Bucket": args.bucket, "Key": args.key}, 1000
    )

    print("Using the Requests package to send a request to the URL.")
    response = None
    if args.action == "get":
        response = requests.get(url)
    elif args.action == "put":
        print("Putting data to the URL.")
        try:
            with open(args.key, "r") as object_file:
```

```

        object_text = object_file.read()
        response = requests.put(url, data=object_text)
    except FileNotFoundError:
        print(
            f"Couldn't find {args.key}. For a PUT operation, the key must be
the "
            f"name of a file that exists on your computer."
        )

    if response is not None:
        print("Got response:")
        print(f"Status: {response.status_code}")
        print(response.text)

    print("-" * 88)

if __name__ == "__main__":
    usage_demo()

```

Generieren Sie eine vorsignierte POST Anfrage zum Hochladen einer Datei.

```

class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
Boto3
        that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name

    def generate_presigned_post(self, object_key, expires_in):
        """
        Generate a presigned Amazon S3 POST request to upload a file.
        A presigned POST can be used for a limited time to let someone without an
AWS
        account upload a file to a bucket.

```

```
:param object_key: The object key to identify the uploaded object.
:param expires_in: The number of seconds the presigned POST is valid.
:return: A dictionary that contains the URL and form fields that contain
        required access data.
"""
try:
    response = self.bucket.meta.client.generate_presigned_post(
        Bucket=self.bucket.name, Key=object_key, ExpiresIn=expires_in
    )
    logger.info("Got presigned POST URL: %s", response["url"])
except ClientError:
    logger.exception(
        "Couldn't get a presigned POST URL for bucket '%s' and object
'%s'",
        self.bucket.name,
        object_key,
    )
    raise
return response
```

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
require "aws-sdk-s3"
require "net/http"

# Creates a presigned URL that can be used to upload content to an object.
#
# @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
# @param object_key [String] The key to give the uploaded object.
# @return [URI, nil] The parsed URI if successful; otherwise nil.
```

```
def get_presigned_url(bucket, object_key)
  url = bucket.object(object_key).presigned_url(:put)
  puts "Created presigned URL: #{url}"
  URI(url)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't create presigned URL for #{bucket.name}:#{object_key}. Here's
  why: #{e.message}"
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-file.txt"
  object_content = "This is the content of my-file.txt."

  bucket = Aws::S3::Bucket.new(bucket_name)
  presigned_url = get_presigned_url(bucket, object_key)
  return unless presigned_url

  response = Net::HTTP.start(presigned_url.host) do |http|
    http.send_request("PUT", presigned_url.request_uri, object_content,
"content_type" => "")
  end

  case response
  when Net::HTTPSuccess
    puts "Content uploaded!"
  else
    puts response.value
  end
end
end

run_demo if $PROGRAM_NAME == __FILE__
```


Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie Presigning-Anfragen für GET S3-Objekte.

```
/// Generate a URL for a presigned GET request.
async fn get_object(
    client: &Client,
    bucket: &str,
    object: &str,
    expires_in: u64,
) -> Result<(), Box<dyn Error>> {
    let expires_in = Duration::from_secs(expires_in);
    let presigned_request = client
        .get_object()
        .bucket(bucket)
        .key(object)
        .presigned(PresigningConfig::expires_in(expires_in)?)
        .await?;

    println!("Object URI: {}", presigned_request.uri());
    let valid_until = chrono::offset::Local::now() + expires_in;
    println!("Valid until: {valid_until}");

    Ok(())
}
```

Erstellen Sie Vorsignierungsanforderungen für PUT S3-Objekte.

```
async fn put_object(
    client: &Client,
    bucket: &str,
    object: &str,
    expires_in: u64,
```

```
) -> Result<String, S3ExampleError> {
    let expires_in: std::time::Duration =
std::time::Duration::from_secs(expires_in);
    let expires_in: aws_sdk_s3::presigning::PresigningConfig =
    PresigningConfig::expires_in(expires_in).map_err(|err| {
        S3ExampleError::new(format!(
            "Failed to convert expiration to PresigningConfig: {err:?}")
        ))
    })?;
    let presigned_request = client
        .put_object()
        .bucket(bucket)
        .key(object)
        .presigned(expires_in)
        .await?;

    Ok(presigned_request.uri().into())
}
```

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Eine Anwendung für Foto-Asset-Management erstellen, mit der Benutzer Fotos mithilfe von Labels verwalten können

Die folgenden Codebeispiele zeigen, wie eine Serverless-Anwendung erstellt wird, mit der Benutzer Fotos mithilfe von Labels verwalten können.

.NET

AWS SDK for .NET

Zeigt, wie eine Anwendung zur Verwaltung von Fotobeständen entwickelt wird, die mithilfe von Amazon Rekognition Labels in Bildern erkennt und sie für einen späteren Abruf speichert.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

Einen ausführlichen Einblick in den Ursprung dieses Beispiels finden Sie im Beitrag auf [AWS Gemeinschaft](#).

In diesem Beispiel verwendete Dienste

- APITor
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

C++

SDK für C++

Zeigt, wie eine Anwendung zur Verwaltung von Fotobeständen entwickelt wird, die mithilfe von Amazon Rekognition Labels in Bildern erkennt und sie für einen späteren Abruf speichert.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

Einen ausführlichen Einblick in den Ursprung dieses Beispiels finden Sie im Beitrag auf [AWS Gemeinschaft](#).

In diesem Beispiel verwendete Dienste

- APITor
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Java

SDK für Java 2.x

Zeigt, wie eine Anwendung zur Verwaltung von Fotobeständen entwickelt wird, die mithilfe von Amazon Rekognition Labels in Bildern erkennt und sie für einen späteren Abruf speichert.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

Einen ausführlichen Einblick in den Ursprung dieses Beispiels finden Sie im Beitrag auf [AWS Gemeinschaft](#).

In diesem Beispiel verwendete Dienste

- APITor
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

JavaScript

SDK für JavaScript (v3)

Zeigt, wie eine Anwendung zur Verwaltung von Fotobeständen entwickelt wird, die mithilfe von Amazon Rekognition Labels in Bildern erkennt und sie für einen späteren Abruf speichert.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

Einen ausführlichen Einblick in den Ursprung dieses Beispiels finden Sie im Beitrag auf [AWS Gemeinschaft](#).

In diesem Beispiel verwendete Dienste

- APITor
- DynamoDB

- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Kotlin

SDK für Kotlin

Zeigt, wie eine Anwendung zur Verwaltung von Fotobeständen entwickelt wird, die mithilfe von Amazon Rekognition Labels in Bildern erkennt und sie für einen späteren Abruf speichert.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

Einen ausführlichen Einblick in den Ursprung dieses Beispiels finden Sie im Beitrag auf [AWS Gemeinschaft](#).

In diesem Beispiel verwendete Dienste

- APITor
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

PHP

SDK für PHP

Zeigt, wie eine Anwendung zur Verwaltung von Fotobeständen entwickelt wird, die mithilfe von Amazon Rekognition Labels in Bildern erkennt und sie für einen späteren Abruf speichert.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

Einen ausführlichen Einblick in den Ursprung dieses Beispiels finden Sie im Beitrag auf [AWS Gemeinschaft](#).

In diesem Beispiel verwendete Dienste

- APITor
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Rust

SDK für Rust

Zeigt, wie eine Anwendung zur Verwaltung von Fotobeständen entwickelt wird, die mithilfe von Amazon Rekognition Labels in Bildern erkennt und sie für einen späteren Abruf speichert.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

Einen ausführlichen Einblick in den Ursprung dieses Beispiels finden Sie im Beitrag auf [AWS Gemeinschaft](#).

In diesem Beispiel verwendete Dienste

- APITor
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Eine Webseite, die Amazon S3 S3-Objekte mithilfe eines auflistet AWS SDK

Im folgenden Codebeispiel wird veranschaulicht, wie Sie Amazon-S3-Objekte auf einer Webseite auflisten.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Der folgende Code ist die relevante React-Komponente, die Aufrufe an AWS SDK. Eine lauffähige Version der Anwendung, die diese Komponente enthält, finden Sie unter dem vorherigen GitHub Link.

```
import { useEffect, useState } from "react";
import {
  ListObjectsCommand,
  ListObjectsCommandOutput,
  S3Client,
} from "@aws-sdk/client-s3";
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import "./App.css";

function App() {
  const [objects, setObjects] = useState<
    Required<ListObjectsCommandOutput>["Contents"]
  >([]);

  useEffect(() => {
    const client = new S3Client({
      region: "us-east-1",
      // Unless you have a public bucket, you'll need access to a private bucket.
      // One way to do this is to create an Amazon Cognito identity pool, attach
      a role to the pool,
      // and grant the role access to the 's3:GetObject' action.
    });
  });
}
```

```
//
// You'll also need to configure the CORS settings on the bucket to allow
traffic from
// this example site. Here's an example configuration that allows all
origins. Don't
// do this in production.
//[
// {
//   "AllowedHeaders": ["*"],
//   "AllowedMethods": ["GET"],
//   "AllowedOrigins": ["*"],
//   "ExposeHeaders": [],
// },
//]
//
credentials: fromCognitoIdentityPool({
  clientConfig: { region: "us-east-1" },
  identityPoolId: "<YOUR_IDENTITY_POOL_ID>",
}),
});
const command = new ListObjectsCommand({ Bucket: "bucket-name" });
client.send(command).then(({ Contents }) => setObjects(Contents || []));
}, []);

return (
  <div className="App">
    {objects.map((o) => (
      <div key={o.ETag}>{o.Key}</div>
    ))}
  </div>
);
}

export default App;
```

- API-Einheiten finden Sie unter [ListObjects](#) AWS SDK for JavaScript API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erstellen Sie eine Amazon-Textract-Explorer-Anwendung

Die folgenden Code-Beispiele zeigen, wie man die Amazon-Textract-Ausgabe in einer interaktiven Anwendung untersuchen kann.

JavaScript

SDK für JavaScript (v3)

Zeigt, wie man das benutzt AWS SDK for JavaScript um eine React-Anwendung zu erstellen, die Amazon Textract verwendet, um Daten aus einem Dokumentbild zu extrahieren und auf einer interaktiven Webseite anzuzeigen. Dieses Beispiel wird in einem Webbrowser ausgeführt und erfordert eine authentifizierte Amazon-Cognito-Identität für Anmeldeinformationen. Es verwendet Amazon Simple Storage Service (Amazon S3) für die Speicherung und für Benachrichtigungen ruft es eine Amazon Simple Queue Service (AmazonSQS) - Warteschlange ab, die ein Amazon Simple Notification Service (AmazonSNS) -Thema abonniert hat.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter. [GitHub](#)

In diesem Beispiel verwendete Dienste

- Amazon Cognito Identity
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Python

SDK für Python (Boto3)

Zeigt, wie man das benutzt AWS SDK for Python (Boto3) mit Amazon Textract zur Erkennung von Text-, Formular- und Tabellenelementen in einem Dokumentbild. Das Eingabe-Image und die Amazon-Textract-Ausgabe werden in einer Tkinter-Anwendung angezeigt, mit der Sie die erkannten Elemente untersuchen können.

- Senden Sie ein Dokument-Image an Amazon Textract und untersuchen Sie die Ausgabe erkannter Elemente.

- Senden Sie Images direkt an Amazon Textract oder über einen Amazon Simple Storage Service (Amazon S3)-Bucket.
- Verwenden Sie asynchronAPIs, um einen Job zu starten, der nach Abschluss des Jobs eine Benachrichtigung zu einem Amazon Simple Notification Service (AmazonSNS) -Thema veröffentlicht.
- Fragen Sie eine Amazon Simple Queue Service (AmazonSQS) -Warteschlange nach einer Auftragsabschlussnachricht ab und zeigen Sie die Ergebnisse an.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Für eine vollständige Liste von AWS SDKEntwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Löschen Sie alle Objekte in einem bestimmten Amazon S3 S3-Bucket mit einem AWS SDK.

Das folgende Codebeispiel zeigt, wie Sie alle Objekte in einem Amazon S3 S3-Bucket löschen.

JavaScript

SDKfür JavaScript (v3)

Note

Es gibt noch mehr dazu [GitHub](#). Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Löscht alle Objekte für einen bestimmten Amazon S3 S3-Bucket.

```
import {
  DeleteObjectsCommand,
  paginateListObjectsV2,
  S3Client,
} from "@aws-sdk/client-s3";

/**
 *
 * @param {{ bucketName: string }} config
 */
export const main = async ({ bucketName }) => {
  const client = new S3Client({});
  try {
    console.log(`Deleting all objects in bucket: ${bucketName}`);

    const paginator = paginateListObjectsV2(
      { client },
      {
        Bucket: bucketName,
      },
    );

    const objectKeys = [];
    for await (const { Contents } of paginator) {
      objectKeys.push(...Contents.map((obj) => ({ Key: obj.Key })));
    }

    const deleteCommand = new DeleteObjectsCommand({
      Bucket: bucketName,
      Delete: { Objects: objectKeys },
    });

    await client.send(deleteCommand);

    console.log(`All objects deleted from bucket: ${bucketName}`);
  } catch (caught) {
    if (caught instanceof Error) {
      console.error(
        `Failed to empty ${bucketName}. ${caught.name}: ${caught.message}`,
      );
    }
  }
}
```

```
};

// Call function if run directly.
import { fileURLToPath } from "url";
import { parseArgs } from "util";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const options = {
    bucketName: {
      type: "string",
    },
  };

  const { values } = parseArgs({ options });
  main(values);
}
```

- APIEinzelheiten finden Sie in den folgenden Themen unter [AWS SDK for JavaScript APIReferenz](#).
 - [DeleteObjects](#)
 - [ListObjectsV2](#)

Für eine vollständige Liste von AWS SDKEntwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Löschen Sie unvollständige mehrteilige Uploads auf Amazon S3 mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie Sie unvollständige mehrteilige Amazon S3 S3-Uploads löschen oder stoppen können.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Um mehrteilige Uploads zu beenden, die gerade bearbeitet werden oder aus irgendeinem Grund unvollständig sind, können Sie eine Liste mit Uploads abrufen und sie dann löschen, wie im folgenden Beispiel gezeigt.

```
public static void abortIncompleteMultipartUploadsFromList() {
    ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
        .bucket(bucketName)
        .build();

    ListMultipartUploadsResponse response =
s3Client.listMultipartUploads(listMultipartUploadsRequest);
    List<MultipartUpload> uploads = response.uploads();

    AbortMultipartUploadRequest abortMultipartUploadRequest;
    for (MultipartUpload upload : uploads) {
        abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
            .bucket(bucketName)
            .key(upload.key())
            .expectedBucketOwner(accountId)
            .uploadId(upload.uploadId())
            .build();

        AbortMultipartUploadResponse abortMultipartUploadResponse =
s3Client.abortMultipartUpload(abortMultipartUploadRequest);
        if (abortMultipartUploadResponse.sdkHttpResponse().isSuccessful()) {
            logger.info("Upload ID [{}] to bucket [{}] successfully
aborted.", upload.uploadId(), bucketName);
        }
    }
}
```

Um unvollständige mehrteilige Uploads zu löschen, die vor oder nach einem Datum initiiert wurden, können Sie mehrteilige Uploads anhand eines bestimmten Zeitpunkts selektiv löschen, wie im folgenden Beispiel gezeigt.

```

static void abortIncompleteMultipartUploadsOlderThan(Instant pointInTime) {
    ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
        .bucket(bucketName)
        .build();

    ListMultipartUploadsResponse response =
s3Client.listMultipartUploads(listMultipartUploadsRequest);
    List<MultipartUpload> uploads = response.uploads();

    AbortMultipartUploadRequest abortMultipartUploadRequest;
    for (MultipartUpload upload : uploads) {
        logger.info("Found multipartUpload with upload ID [{}], initiated
[{}]", upload.uploadId(), upload.initiated());
        if (upload.initiated().isBefore(pointInTime)) {
            abortMultipartUploadRequest =
AbortMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(upload.key())
                .expectedBucketOwner(accountId)
                .uploadId(upload.uploadId())
                .build();

            AbortMultipartUploadResponse abortMultipartUploadResponse =
s3Client.abortMultipartUpload(abortMultipartUploadRequest);
            if
(abortMultipartUploadResponse.sdkHttpResponse().isSuccessful()) {
                logger.info("Upload ID [{}] to bucket [{}] successfully
aborted.", upload.uploadId(), bucketName);
            }
        }
    }
}

```

Wenn Sie nach dem Start eines mehrteiligen Uploads Zugriff auf die Upload-ID haben, können Sie den laufenden Upload mithilfe der ID löschen.

```

static void abortMultipartUploadUsingUploadId() {

```

```

String uploadId = startUploadReturningUploadId();
AbortMultipartUploadResponse response = s3Client.abortMultipartUpload(b -
> b
    .uploadId(uploadId)
    .bucket(bucketName)
    .key(key));

if (response.sdkHttpResponse().isSuccessful()) {
    logger.info("Upload ID [{}] to bucket [{}] successfully aborted.",
uploadId, bucketName);
}
}

```

Um unvollständige mehrteilige Uploads, die älter als eine bestimmte Anzahl von Tagen sind, dauerhaft zu löschen, richten Sie eine Bucket-Lebenszykluskonfiguration für den Bucket ein. Das folgende Beispiel zeigt, wie Sie eine Regel zum Löschen unvollständiger Uploads erstellen, die älter als 7 Tage sind.

```

static void abortMultipartUploadsUsingLifecycleConfig() {
    Collection<LifecycleRule> lifeCycleRules =
List.of(LifecycleRule.builder()
    .abortIncompleteMultipartUpload(b -> b.
        daysAfterInitiation(7))
    .status("Enabled")
    .filter(SdkBuilder::build) // Filter element is required.
    .build());

    // If the action is successful, the service sends back an HTTP 200
response with an empty HTTP body.
    PutBucketLifecycleConfigurationResponse response =
s3Client.putBucketLifecycleConfiguration(b -> b
    .bucket(bucketName)
    .lifecycleConfiguration(b1 -> b1.rules(lifeCycleRules)));

    if (response.sdkHttpResponse().isSuccessful()) {
        logger.info("Rule to abort incomplete multipart uploads added to
bucket.");
    } else {
        logger.error("Unsuccessfully applied rule. HTTP status code is [{}]",
response.sdkHttpResponse().statusCode());
    }
}
}

```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for Java 2.x API-Referenz](#).
 - [AbortMultipartUpload](#)
 - [ListMultipartUploads](#)
 - [PutBucketLifecycleConfiguration](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erkennung PPE in Bildern mit Amazon Rekognition mithilfe eines AWS SDK

Das folgende Codebeispiel zeigt, wie Sie eine App erstellen, die Amazon Rekognition verwendet, um persönliche Schutzausrüstung (PPE) in Bildern zu erkennen.

Java

SDK für Java 2.x

Zeigt, wie man eine erstellte AWS Lambda-Funktion, die Bilder mit persönlicher Schutzausrüstung erkennt.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erkennen Sie Entitäten in Text, der aus einem Bild extrahiert wurde, mithilfe eines AWS SDK

Das folgende Codebeispiel zeigt, wie Amazon Comprehend verwendet wird, um Entitäten in Text zu erkennen, der von Amazon Textract aus einem in Amazon S3 gespeicherten Bild extrahiert wurde.

Python

SDK für Python (Boto3)

Zeigt, wie man das benutzt AWS SDK for Python (Boto3) in einem Jupyter-Notizbuch, um Entitäten in Text zu erkennen, der aus einem Bild extrahiert wurde. In diesem Beispiel extrahiert Amazon Textract Text aus einem Bild, das in Amazon Simple Storage Service (Amazon S3) und Amazon Comprehend gespeichert ist, um Entitäten im extrahierten Text zu erkennen.

Dieses Beispiel ist ein Jupyter Notebook und muss in einer Umgebung ausgeführt werden, die Notebooks hosten kann. Anweisungen zur Ausführung des Beispiels mit Amazon SageMaker finden Sie in den Anweisungen in der Datei [TextractAndComprehendNotebook.ipynb](#).

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon S3
- Amazon Textract

Eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erkennen Sie Gesichter in einem Bild mit einem AWS SDK

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Speichern Sie ein Bild in einem Amazon-S3-Bucket.

- Verwenden Sie Amazon Rekognition, um Gesichtsdetails wie Altersgruppe, Geschlecht und Emotionen (z B. Lächeln) zu erkennen.
- Zeigen Sie diese Details an.

Rust

SDK für Rust

Speichern Sie das Bild in einem Amazon-S3-Bucket mit einem uploads-Präfix, verwenden Sie Amazon Rekognition, um Gesichtsdetails wie Altersgruppe, Geschlecht und Emotionen (Lächeln usw.) zu erkennen, und zeigen Sie diese Details an.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Rekognition
- Amazon S3

Eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erkennen Sie Objekte in Bildern mit Amazon Rekognition mithilfe eines AWS SDK

Die folgenden Code-Beispiele zeigen, wie man eine App erstellt, die Amazon Rekognition verwendet, um Objekte nach Kategorien in Bildern zu erkennen.

.NET

AWS SDK for .NET

Zeigt, wie Amazon Rekognition verwendet wird. NETAPI um eine App zu erstellen, die Amazon Rekognition verwendet, um Objekte in Bildern, die sich in einem Amazon Simple Storage Service (Amazon S3) -Bucket befinden, nach Kategorien zu identifizieren. Die App sendet dem Administrator mithilfe von Amazon Simple Email Service (AmazonSES) eine E-Mail-Benachrichtigung mit den Ergebnissen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Rekognition
- Amazon S3
- Amazon SES

Java

SDK für Java 2.x

Zeigt, wie Amazon Rekognition Java verwendet wird, um eine App API zu erstellen, die Amazon Rekognition verwendet, um Objekte nach Kategorien in Bildern zu identifizieren, die sich in einem Amazon Simple Storage Service (Amazon S3) -Bucket befinden. Die App sendet dem Administrator mithilfe von Amazon Simple Email Service (AmazonSES) eine E-Mail-Benachrichtigung mit den Ergebnissen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK für JavaScript (v3)

Zeigt, wie Sie Amazon Rekognition mit dem verwenden AWS SDK for JavaScript um eine App zu erstellen, die Amazon Rekognition verwendet, um Objekte in Bildern, die sich in einem Amazon Simple Storage Service (Amazon S3) -Bucket befinden, nach Kategorien zu identifizieren. Die App sendet dem Administrator mithilfe von Amazon Simple Email Service (AmazonSES) eine E-Mail-Benachrichtigung mit den Ergebnissen.

So funktioniert es:

- Erstellen Sie mit Amazon Cognito einen nicht authentifizierten Benutzer.

- Analysieren Sie mit Amazon Rekognition Bilder für Objekte.
- Bestätigen Sie eine E-Mail-Adresse für AmazonSES.
- Senden Sie eine E-Mail-Benachrichtigung über AmazonSES.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Rekognition
- Amazon S3
- Amazon SES

Kotlin

SDK für Kotlin

Zeigt, wie Amazon Rekognition Kotlin verwendet wird, um eine App API zu erstellen, die Amazon Rekognition verwendet, um Objekte nach Kategorien in Bildern zu identifizieren, die sich in einem Amazon Simple Storage Service (Amazon S3) -Bucket befinden. Die App sendet dem Administrator mithilfe von Amazon Simple Email Service (AmazonSES) eine E-Mail-Benachrichtigung mit den Ergebnissen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Rekognition
- Amazon S3
- Amazon SES

Python

SDK für Python (Boto3)

Zeigt dir, wie man das benutzt AWS SDK for Python (Boto3) um eine Webanwendung zu erstellen, mit der Sie Folgendes tun können:

- Laden Sie Fotos in einen Bucket von Amazon Simple Storage Service (Amazon S3) hoch.

- Verwenden Sie Amazon Rekognition, um die Fotos zu analysieren und zu markieren.
- Verwenden Sie Amazon Simple Email Service (AmazonSES), um E-Mail-Berichte zur Bildanalyse zu versenden.

Dieses Beispiel enthält zwei Hauptkomponenten: eine eingeschriebene Webseite JavaScript, die mit React erstellt wurde, und einen in Python geschriebenen REST Dienst, der mit Flask-RESTful erstellt wurde.

Sie können die React-Webseite verwenden, um Folgendes auszuführen:

- Zeigen Sie eine Liste der Bilder an, die in Ihrem S3-Bucket gespeichert sind.
- Laden Sie Bilder von Ihrem Computer in Ihren S3-Bucket hoch.
- Zeigen Sie Bilder und Markierungen an, die Elemente identifizieren, welche im Bild erkannt werden.
- Rufen Sie einen Bericht über alle Bilder in Ihrem S3-Bucket ab und senden Sie eine E-Mail mit dem Bericht.

Die Webseite ruft den REST Dienst auf. Der Dienst sendet Anfragen an AWS um die folgenden Aktionen durchzuführen:

- Die Liste der Bilder abrufen und in Ihrem S3-Bucket filtern.
- Fotos in Ihren S3-Bucket hochladen.
- Verwenden Sie Amazon Rekognition, um einzelne Fotos zu analysieren und eine Liste von Markierungen zu erhalten, die die auf dem Foto erkannten Elemente identifizieren.
- Analysieren Sie alle Fotos in Ihrem S3-Bucket und verwenden Sie AmazonSES, um einen Bericht per E-Mail zu versenden.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Rekognition
- Amazon S3
- Amazon SES

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erkennen Sie Personen und Objekte in einem Video mit Amazon Rekognition mithilfe eines AWS SDK

Die folgenden Code-Beispiele zeigen, wie man Personen und Objekte in einem Video mit Amazon Rekognition erkennt.

Java

SDK für Java 2.x

Zeigt, wie Amazon Rekognition Java verwendet wird, API um eine App zur Erkennung von Gesichtern und Objekten in Videos zu erstellen, die sich in einem Amazon Simple Storage Service (Amazon S3) -Bucket befinden. Die App sendet dem Administrator mithilfe von Amazon Simple Email Service (AmazonSES) eine E-Mail-Benachrichtigung mit den Ergebnissen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Rekognition
- Amazon S3
- Amazon SES

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Herunterladen aller Objekte aus einem Amazon Simple Storage Service (Amazon S3)-Bucket in ein lokales Verzeichnis

Das folgende Codebeispiel zeigt, wie Sie alle Objekte aus einem Amazon Simple Storage Service (Amazon S3)-Bucket in ein lokales Verzeichnis herunterladen.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Verwenden Sie ein [S3 TransferManager](#), um [alle S3-Objekte im selben S3-Bucket herunterzuladen](#). Sehen Sie sich die [vollständige Datei](#) an und [testen](#) Sie sie.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadDirectoryRequest;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.HashSet;
import java.util.Set;
import java.util.UUID;
import java.util.stream.Collectors;

public Integer downloadObjectsToDirectory(S3TransferManager transferManager,
    URI destinationPathURI, String bucketName) {
    DirectoryDownload directoryDownload =
transferManager.downloadDirectory(DownloadDirectoryRequest.builder()
        .destination(Paths.get(destinationPathURI))
        .bucket(bucketName)
        .build());
    CompletedDirectoryDownload completedDirectoryDownload =
directoryDownload.completionFuture().join();
```

```
        completedDirectoryDownload.failedTransfers()
            .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
        return completedDirectoryDownload.failedTransfers().size();
    }
}
```

- API-Einheiten finden Sie [DownloadDirectory](#) unter AWS SDK for Java 2.x API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Rufen Sie ein Amazon S3 S3-Objekt von einem Access Point mit mehreren Regionen ab, indem Sie eine AWS SDK

Das folgende Codebeispiel zeigt, wie Sie ein Objekt von einem Access Point mit mehreren Regionen abrufen.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Konfigurieren Sie den S3-Client so, dass er den Asymmetric Sigv4 (Sigv4A) - Signaturalgorithmus verwendet.

```
suspend fun createS3Client(): S3Client {
    // Configure your S3Client to use the Asymmetric Sigv4 (Sigv4a)
    signing algorithm.
    val sigV4AScheme = SigV4AsymmetricAuthScheme(CrtAwsSigner)
    val s3 = S3Client.fromEnvironment {
        authSchemes = listOf(sigV4AScheme)
    }
    return s3
}
```



```
}
```

Verwenden Sie den Multi-Region Access Point ARN anstelle eines Bucket-Namens, um das Objekt abzurufen.

```
suspend fun getObjectFromMrap(
    s3: S3Client,
    mrapArn: String,
    keyName: String,
): String? {
    val request = GetObjectRequest {
        bucket = mrapArn // Use the ARN instead of the bucket name for object
operations.
        key = keyName
    }

    var stringObj: String? = null
    s3.getObject(request) { resp ->
        stringObj = resp.body?.decodeToString()
        if (stringObj != null) {
            println("Successfully read $keyName from $mrapArn")
        }
    }
    return stringObj
}
```

- Weitere Informationen finden Sie unter [AWS SDK für den Kotlin-Entwicklerhandbuch](#).
- API-Einheiten finden Sie in [GetObject](#) AWS SDK als API Kotlin-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Holen Sie sich ein Objekt aus einem Amazon S3 S3-Bucket mit einem AWS SDK, unter Angabe eines If-Modified-Since-Headers

Das folgende Codebeispiel zeigt, wie Sie Daten aus einem Objekt in einem S3-Bucket lesen, jedoch nur, wenn dieser Bucket seit dem letzten Abruf nicht geändert wurde.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
use aws_sdk_s3::{
    error::SdkError,
    primitives::{ByteStream, DateTime, DateTimeFormat},
    Client,
};
use s3_code_examples::error::S3ExampleError;
use tracing::{error, warn};

const KEY: &str = "key";
const BODY: &str = "Hello, world!";

/// Demonstrate how `if-modified-since` reports that matching objects haven't
/// changed.
///
/// # Steps
/// - Create a bucket.
/// - Put an object in the bucket.
/// - Get the bucket headers.
/// - Get the bucket headers again but only if modified.
/// - Delete the bucket.
#[tokio::main]
async fn main() -> Result<(), S3ExampleError> {
    tracing_subscriber::fmt::init();

    // Get a new UUID to use when creating a unique bucket name.
    let uuid = uuid::Uuid::new_v4();

    // Load the AWS configuration from the environment.
    let client = Client::new(&aws_config::load_from_env().await);

    // Generate a unique bucket name using the previously generated UUID.
    // Then create a new bucket with that name.
```

```
let bucket_name = format!("if-modified-since-{{uuid}}");
client
    .create_bucket()
    .bucket(bucket_name.clone())
    .send()
    .await?;

// Create a new object in the bucket whose name is `KEY` and whose
// contents are `BODY`.
let put_object_output = client
    .put_object()
    .bucket(bucket_name.as_str())
    .key(KEY)
    .body(ByteStream::from_static(BODY.as_bytes()))
    .send()
    .await;

// If the `PutObject` succeeded, get the eTag string from it. Otherwise,
// report an error and return an empty string.
let e_tag_1 = match put_object_output {
    Ok(put_object) => put_object.e_tag.unwrap(),
    Err(err) => {
        error!("{err:?}");
        String::new()
    }
};

// Request the object's headers.
let head_object_output = client
    .head_object()
    .bucket(bucket_name.as_str())
    .key(KEY)
    .send()
    .await;

// If the `HeadObject` request succeeded, create a tuple containing the
// values of the headers `last-modified` and `etag`. If the request
// failed, return the error in a tuple instead.
let (last_modified, e_tag_2) = match head_object_output {
    Ok(head_object) => (
        Ok(head_object.last_modified().cloned().unwrap()),
        head_object.e_tag.unwrap(),
    ),
    Err(err) => (Err(err), String::new()),
};
```

```
};

warn!("last modified: {last_modified:?}");
assert_eq!(
    e_tag_1, e_tag_2,
    "PutObject and first GetObject had differing eTags"
);

println!("First value of last_modified: {last_modified:?}");
println!("First tag: {}\n", e_tag_1);

// Send a second `HeadObject` request. This time, the `if_modified_since`
// option is specified, giving the `last_modified` value returned by the
// first call to `HeadObject`.
//
// Since the object hasn't been changed, and there are no other objects in
// the bucket, there should be no matching objects.

let head_object_output = client
    .head_object()
    .bucket(bucket_name.as_str())
    .key(KEY)
    .if_modified_since(last_modified.unwrap())
    .send()
    .await;

// If the `HeadObject` request succeeded, the result is a tuple containing
// the `last_modified` and `e_tag_1` properties. This is not the expected
// result.
//
// The expected result of the second call to `HeadObject` is an
// `SdkError::ServiceError` containing the HTTP error response. If that's
// the case and the HTTP status is 304 (not modified), the output is a
// tuple containing the values of the HTTP `last-modified` and `etag`
// headers.
//
// If any other HTTP error occurred, the error is returned as an
// `SdkError::ServiceError`.

let (last_modified, e_tag_2) = match head_object_output {
    Ok(head_object) => (
        Ok(head_object.last_modified().cloned().unwrap()),
        head_object.e_tag.unwrap(),
    ),
```

```

Err(err) => match err {
  SdkError::ServiceError(err) => {
    // Get the raw HTTP response. If its status is 304, the
    // object has not changed. This is the expected code path.
    let http = err.raw();
    match http.status().as_u16() {
      // If the HTTP status is 304: Not Modified, return a
      // tuple containing the values of the HTTP
      // `last-modified` and `etag` headers.
      304 => (
        Ok(DateTime::from_str(
          http.headers().get("last-modified").unwrap(),
          DateTimeFormat::HttpDate,
        )
        .unwrap()),
        http.headers().get("etag").map(|t| t.into()).unwrap(),
      ),
      // Any other HTTP status code is returned as an
      // `SdkError::ServiceError`.
      _ => (Err(SdkError::ServiceError(err)), String::new()),
    }
  }
  // Any other kind of error is returned in a tuple containing the
  // error and an empty string.
  _ => (Err(err), String::new()),
},
};

warn!("last modified: {last_modified:?}");
assert_eq!(
  e_tag_1, e_tag_2,
  "PutObject and second HeadObject had different eTags"
);

println!("Second value of last modified: {last_modified:?}");
println!("Second tag: {}", e_tag_2);

// Clean up by deleting the object and the bucket.
client
  .delete_object()
  .bucket(bucket_name.as_str())
  .key(KEY)
  .send()
  .await?;

```

```
client
    .delete_bucket()
    .bucket(bucket_name.as_str())
    .send()
    .await?;

Ok(())
}
```

- API-Einheiten finden Sie [GetObject](#) unter AWS SDK als API Referenz auf Rust.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte mit der Verschlüsselung von Amazon S3 S3-Objekten mithilfe eines AWS SDK

Das folgende Codebeispiel veranschaulicht die ersten Schritte mit der Verschlüsselung von Amazon-S3-Objekten.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
```

```
/// <summary>
/// This example shows how to apply client encryption to an object in an
/// Amazon Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class SSEClientEncryption
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";
        string keyName = "exampleobject.txt";
        string copyTargetKeyName = "examplecopy.txt";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2.
        IAmazonS3 client = new AmazonS3Client();

        try
        {
            // Create an encryption key.
            Aes aesEncryption = Aes.Create();
            aesEncryption.KeySize = 256;
            aesEncryption.GenerateKey();
            string base64Key = Convert.ToBase64String(aesEncryption.Key);

            // Upload the object.
            PutObjectRequest putObjectRequest = await
UploadObjectAsync(client, bucketName, keyName, base64Key);

            // Download the object and verify that its contents match what
you uploaded.
            await DownloadObjectAsync(client, bucketName, keyName, base64Key,
putObjectRequest);

            // Get object metadata and verify that the object uses AES-256
encryption.
            await GetObjectMetadataAsync(client, bucketName, keyName,
base64Key);

            // Copy both the source and target objects using server-side
encryption with
            // an encryption key.
```

```

        await CopyObjectAsync(client, bucketName, keyName,
copyTargetKeyName, aesEncryption, base64Key);
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

/// <summary>
/// Uploads an object to an Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used to
call
the
S3
/// PutObjectAsync.</param>
/// <param name="bucketName">The name of the Amazon S3 bucket to which
the
S3
/// object will be uploaded.</param>
/// <param name="keyName">The name of the object to upload to the Amazon
S3
/// bucket.</param>
/// <param name="base64Key">The encryption key.</param>
/// <returns>The PutObjectRequest object for use by
DownloadObjectAsync.</returns>
public static async Task<PutObjectRequest> UploadObjectAsync(
    IAmazonS3 client,
    string bucketName,
    string keyName,
    string base64Key)
{
    PutObjectRequest putObjectRequest = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = keyName,
        ContentBody = "sample text",
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key,
    };
    PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
    return putObjectRequest;
}

```



```
    /// <summary>
    /// Downloads an encrypted object from an Amazon S3 bucket.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// GetObjectAsync.</param>
    /// <param name="bucketName">The name of the Amazon S3 bucket where the
object
    /// is located.</param>
    /// <param name="keyName">The name of the Amazon S3 object to download.</
param>
    /// <param name="base64Key">The encryption key used to encrypt the
    /// object.</param>
    /// <param name="putObjectRequest">The PutObjectRequest used to upload
    /// the object.</param>
    public static async Task DownloadObjectAsync(
        IAmazonS3 client,
        string bucketName,
        string keyName,
        string base64Key,
        PutObjectRequest putObjectRequest)
    {
        GetObjectRequest getObjectRequest = new GetObjectRequest
        {
            BucketName = bucketName,
            Key = keyName,

            // Provide encryption information for the object stored in Amazon
S3.
            ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
            ServerSideEncryptionCustomerProvidedKey = base64Key,
        };

        using (GetObjectResponse getResponse = await
client.GetObjectAsync(getObjectRequest))
            using (StreamReader reader = new
StreamReader(getResponse.ResponseStream))
            {
                string content = reader.ReadToEnd();
                if (string.Compare(putObjectRequest.ContentBody, content) == 0)
                {
                    Console.WriteLine("Object content is same as we uploaded");
                }
            }
    }
}
```

```
        else
        {
            Console.WriteLine("Error...Object content is not same.");
        }

        if (getResponse.ServerSideEncryptionCustomerMethod ==
ServerSideEncryptionCustomerMethod.AES256)
        {
            Console.WriteLine("Object encryption method is AES256, same
as we set");
        }
        else
        {
            Console.WriteLine("Error...Object encryption method is not
the same as AES256 we set");
        }
    }
}

/// <summary>
/// Retrieves the metadata associated with an Amazon S3 object.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used
/// to call GetObjectMetadataAsync.</param>
/// <param name="bucketName">The name of the Amazon S3 bucket containing
the
/// object for which we want to retrieve metadata.</param>
/// <param name="keyName">The name of the object for which we wish to
/// retrieve the metadata.</param>
/// <param name="base64Key">The encryption key associated with the
/// object.</param>
public static async Task GetObjectMetadataAsync(
    IAmazonS3 client,
    string bucketName,
    string keyName,
    string base64Key)
{
    GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest
    {
        BucketName = bucketName,
        Key = keyName,
```

```
        // The object stored in Amazon S3 is encrypted, so provide the
        necessary encryption information.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = base64Key,
    };

    GetObjectMetadataResponse getObjectMetadataResponse = await
client.GetObjectMetadataAsync(getObjectMetadataRequest);
    Console.WriteLine("The object metadata show encryption method used
is: {0}", getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
    }

    /// <summary>
    /// Copies an encrypted object from one Amazon S3 bucket to another.
    /// </summary>
    /// <param name="client">The initialized Amazon S3 client object used to
call
    /// CopyObjectAsync.</param>
    /// <param name="bucketName">The Amazon S3 bucket containing the object
    /// to copy.</param>
    /// <param name="keyName">The name of the object to copy.</param>
    /// <param name="copyTargetKeyName">The Amazon S3 bucket to which the
object
    /// will be copied.</param>
    /// <param name="aesEncryption">The encryption type to use.</param>
    /// <param name="base64Key">The encryption key to use.</param>
    public static async Task CopyObjectAsync(
        IAmazonS3 client,
        string bucketName,
        string keyName,
        string copyTargetKeyName,
        Aes aesEncryption,
        string base64Key)
    {
        aesEncryption.GenerateKey();
        string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);

        CopyObjectRequest copyRequest = new CopyObjectRequest
        {
            SourceBucket = bucketName,
            SourceKey = keyName,
            DestinationBucket = bucketName,
            DestinationKey = copyTargetKeyName,
```

```
        // Information about the source object's encryption.
        CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,

        // Information about the target object's encryption.
        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
        ServerSideEncryptionCustomerProvidedKey = copyBase64Key,
    };
    await client.CopyObjectAsync(copyRequest);
}
}
```

- APIEinzelheiten finden Sie in den folgenden Themen unter AWS SDK for .NET APIReferenz.
 - [CopyObject](#)
 - [GetObject](#)
 - [GetObjectMetadata](#)

Für eine vollständige Liste von AWS SDKEntwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Erste Schritte mit Tags für Amazon S3 S3-Objekte mithilfe eines AWS SDK

Das folgende Codebeispiel zeigt die ersten Schritte mit Tags für Amazon-S3-Objekte.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to work with tags in Amazon Simple Storage
/// Service (Amazon S3) objects.
/// </summary>
public class ObjectTag
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";
        string keyName = "newobject.txt";
        string filePath = @"*** file path ***";

        // Specify your bucket region (an example region is shown).
        RegionEndpoint bucketRegion = RegionEndpoint.USWest2;

        var client = new AmazonS3Client(bucketRegion);
        await PutObjectsWithTagsAsync(client, bucketName, keyName, filePath);
    }

    /// <summary>
    /// This method uploads an object with tags. It then shows the tag
    /// values, changes the tags, and shows the new tags.
    /// </summary>
    /// <param name="client">The Initialized Amazon S3 client object used
    /// to call the methods to create and change an objects tags.</param>
    /// <param name="bucketName">A string representing the name of the
    /// bucket where the object will be stored.</param>
    /// <param name="keyName">A string representing the key name of the
    /// object to be tagged.</param>
    /// <param name="filePath">The directory location and file name of the
    /// object to be uploaded to the Amazon S3 bucket.</param>
    public static async Task PutObjectsWithTagsAsync(IAmazonS3 client, string
bucketName, string keyName, string filePath)
    {
        try
        {
```

```
// Create an object with tags.
var putRequest = new PutObjectRequest
{
    BucketName = bucketName,
    Key = keyName,
    FilePath = filePath,
    TagSet = new List<Tag>
    {
        new Tag { Key = "Keyx1", Value = "Value1" },
        new Tag { Key = "Keyx2", Value = "Value2" },
    },
};

PutObjectResponse response = await
client.PutObjectAsync(putRequest);

// Now retrieve the new object's tags.
GetObjectTaggingRequest getTagsRequest = new
GetObjectTaggingRequest()
{
    BucketName = bucketName,
    Key = keyName,
};

GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);

// Display the tag values.
objectTags.Tagging
    .ForEach(t => Console.WriteLine($"Key: {t.Key}, Value:
{t.Value}"));

Tagging newTagSet = new Tagging()
{
    TagSet = new List<Tag>
    {
        new Tag { Key = "Key3", Value = "Value3" },
        new Tag { Key = "Key4", Value = "Value4" },
    },
};

PutObjectTaggingRequest putObjTagsRequest = new
PutObjectTaggingRequest()
{
```

```
        BucketName = bucketName,
        Key = keyName,
        Tagging = newTagSet,
    };

    PutObjectTaggingResponse response2 = await
client.PutObjectTaggingAsync(putObjTagsRequest);

    // Retrieve the tags again and show the values.
    GetObjectTaggingRequest getTagsRequest2 = new
GetObjectTaggingRequest()
    {
        BucketName = bucketName,
        Key = keyName,
    };
    GetObjectTaggingResponse objectTags2 = await
client.GetObjectTaggingAsync(getTagsRequest2);

    objectTags2.Tagging
        .ForEach(t => Console.WriteLine($"Key: {t.Key}, Value:
{t.Value}"));
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine(
            $"Error: '{ex.Message}'");
    }
}
}
```

- API-Einheiten finden Sie [GetObjectTagging](#) unter AWS SDK for .NET API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Arbeiten Sie mit den Objektsperreffunktionen von Amazon S3 mithilfe eines AWS SDK

Die folgenden Codebeispiele zeigen, wie Sie mit S3-Objektsperreffunktionen arbeiten.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Führen Sie ein interaktives Szenario aus, in dem die Objektsperreffunktionen von Amazon S3 demonstriert werden.

```
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Logging.Console;
using Microsoft.Extensions.Logging.Debug;

namespace S3ObjectLockScenario;

public static class S3ObjectLockWorkflow
{
    /*
        Before running this .NET code example, set up your development environment,
        including your credentials.

        This .NET example performs the following tasks:
        1. Create test Amazon Simple Storage Service (S3) buckets with different
        lock policies.
        2. Upload sample objects to each bucket.
        3. Set some Legal Hold and Retention Periods on objects and buckets.
        4. Investigate lock policies by viewing settings or attempting to delete
        or overwrite objects.
        5. Clean up objects and buckets.
    */

    public static S3ActionsWrapper _s3ActionsWrapper = null!;
```



```
public static IConfiguration _configuration = null!;  
private static string _resourcePrefix = null!;  
private static string noLockBucketName = null!;  
private static string lockEnabledBucketName = null!;  
private static string retentionAfterCreationBucketName = null!;  
private static List<string> bucketNames = new List<string>();  
private static List<string> fileNames = new List<string>();  
  
public static async Task Main(string[] args)  
{  
    // Set up dependency injection for the Amazon service.  
    using var host = Host.CreateDefaultBuilder(args)  
        .ConfigureLogging(logging =>  
            logging.AddFilter("System", LogLevel.Debug)  
                .AddFilter<DebugLoggerProvider>("Microsoft",  
LogLevel.Information)  
                .AddFilter<ConsoleLoggerProvider>("Microsoft",  
LogLevel.Trace))  
        .ConfigureServices((_, services) =>  
            services.AddAWSService<IAmazonS3>()  
                .AddTransient<S3ActionsWrapper>()  
        )  
        .Build();  
  
    _configuration = new ConfigurationBuilder()  
        .SetBasePath(Directory.GetCurrentDirectory())  
        .AddJsonFile("settings.json") // Load settings from .json file.  
        .AddJsonFile("settings.local.json",  
            true) // Optionally, load local settings.  
        .Build();  
  
    ConfigurationSetup();  
  
    ServicesSetup(host);  
  
    try  
    {  
        Console.WriteLine(new string('-', 80));  
        Console.WriteLine("Welcome to the Amazon Simple Storage Service (S3)  
Object Locking Workflow Scenario.");  
        Console.WriteLine(new string('-', 80));  
        await Setup(true);  
  
        await DemoActionChoices();  
    }  
}
```

```
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Cleaning up resources.");
        Console.WriteLine(new string('-', 80));
        await Cleanup(true);

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Amazon S3 Object Locking Workflow is complete.");
        Console.WriteLine(new string('-', 80));
    }
    catch (Exception ex)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"There was a problem: {ex.Message}");
        await Cleanup(true);
        Console.WriteLine(new string('-', 80));
    }
}

/// <summary>
/// Populate the services for use within the console application.
/// </summary>
/// <param name="host">The services host.</param>
private static void ServicesSetup(IHost host)
{
    _s3ActionsWrapper = host.Services.GetRequiredService<S3ActionsWrapper>();
}

/// <summary>
/// Any setup operations needed.
/// </summary>
public static void ConfigurationSetup()
{
    _resourcePrefix = _configuration["resourcePrefix"] ?? "dotnet-example";

    noLockBucketName = _resourcePrefix + "-no-lock";
    lockEnabledBucketName = _resourcePrefix + "-lock-enabled";
    retentionAfterCreationBucketName = _resourcePrefix + "-retention-after-
creation";

    bucketNames.Add(noLockBucketName);
    bucketNames.Add(lockEnabledBucketName);
    bucketNames.Add(retentionAfterCreationBucketName);
}
}
```

```
// <summary>
/// Deploy necessary resources for the scenario.
/// </summary>
/// <param name="interactive">True to run as interactive.</param>
/// <returns>True if successful.</returns>
public static async Task<bool> Setup(bool interactive)
{
    Console.WriteLine(
        "\nFor this workflow, we will use the AWS SDK for .NET to create
several S3\n" +
        "buckets and files to demonstrate working with S3 locking features.
\n");

    Console.WriteLine(new string('-', 80));
    Console.WriteLine("Press Enter when you are ready to start.");
    if (interactive)
        Console.ReadLine();

    Console.WriteLine("\nS3 buckets can be created either with or without
object lock enabled.");
    await _s3ActionsWrapper.CreateBucketWithObjectLock(noLockBucketName,
false);
    await _s3ActionsWrapper.CreateBucketWithObjectLock(lockEnabledBucketName,
true);
    await
_s3ActionsWrapper.CreateBucketWithObjectLock(retentionAfterCreationBucketName,
false);

    Console.WriteLine("Press Enter to continue.");
    if (interactive)
        Console.ReadLine();

    Console.WriteLine("\nA bucket can be configured to use object locking
with a default retention period.");
    await
_s3ActionsWrapper.ModifyBucketDefaultRetention(retentionAfterCreationBucketName,
true,
        ObjectLockRetentionMode.Governance, DateTime.UtcNow.AddDays(1));

    Console.WriteLine("Press Enter to continue.");
    if (interactive)
        Console.ReadLine();
}
```

```
    Console.WriteLine("\nObject lock policies can also be added to existing
buckets.");
    await _s3ActionsWrapper.EnableObjectLockOnBucket(lockEnabledBucketName);

    Console.WriteLine("Press Enter to continue.");
    if (interactive)
        Console.ReadLine();

    // Upload some files to the buckets.
    Console.WriteLine("\nNow let's add some test files:");
    var fileName = _configuration["exampleFileName"] ?? "exampleFile.txt";
    int fileCount = 2;
    // Create the file if it does not already exist.
    if (!File.Exists(fileName))
    {
        await using StreamWriter sw = File.CreateText(fileName);
        await sw.WriteLineAsync(
            "This is a sample file for uploading to a bucket.");
    }

    foreach (var bucketName in bucketNames)
    {
        for (int i = 0; i < fileCount; i++)
        {
            var numberedFileName = Path.GetFileNameWithoutExtension(fileName)
+ i + Path.GetExtension(fileName);
            fileNames.Add(numberedFileName);
            await _s3ActionsWrapper.UploadFileAsync(bucketName,
numberedFileName, fileName);
        }
    }
    Console.WriteLine("Press Enter to continue.");
    if (interactive)
        Console.ReadLine();

    if (!interactive)
        return true;
    Console.WriteLine("\nNow we can set some object lock policies on
individual files:");
    foreach (var bucketName in bucketNames)
    {
        for (int i = 0; i < fileNames.Count; i++)
        {
            // No modifications to the objects in the first bucket.
```

```

        if (bucketName != bucketNames[0])
        {
            var exampleFileName = fileNames[i];
            switch (i)
            {
                case 0:
                {
                    var question =
                        $"Would you like to add a legal hold to
{exampleFileName} in {bucketName}? (y/n)";
                    if (GetYesNoResponse(question))
                    {
                        // Set a legal hold.
                        await
_s3ActionsWrapper.ModifyObjectLegalHold(bucketName, exampleFileName,
ObjectLockLegalHoldStatus.On);

                    }
                    break;
                }
                case 1:
                {
                    var question =
                        $"Would you like to add a 1 day Governance
retention period to {exampleFileName} in {bucketName}? (y/n)" +
                        "\nReminder: Only a user with the
s3:BypassGovernanceRetention permission will be able to delete this file or its
bucket until the retention period has expired.";
                    if (GetYesNoResponse(question))
                    {
                        // Set a Governance mode retention period for
1 day.
                        await
_s3ActionsWrapper.ModifyObjectRetentionPeriod(
                            bucketName, exampleFileName,
                            ObjectLockRetentionMode.Governance,
                            DateTime.UtcNow.AddDays(1));
                    }
                    break;
                }
            }
        }
    }
}

```

```
        Console.WriteLine(new string('-', 80));
        return true;
    }

    /// <summary>
    /// List all of the current buckets and objects.
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>The list of buckets and objects.</returns>
    public static async Task<List<S3ObjectVersion>> ListBucketsAndObjects(bool
interactive)
    {
        var allObjects = new List<S3ObjectVersion>();
        foreach (var bucketName in bucketNames)
        {
            var objectsInBucket = await
_s3ActionsWrapper.ListBucketObjectsAndVersions(bucketName);
            foreach (var objectKey in objectsInBucket.Versions)
            {
                allObjects.Add(objectKey);
            }
        }

        if (interactive)
        {
            Console.WriteLine("\nCurrent buckets and objects:\n");
            int i = 0;
            foreach (var bucketObject in allObjects)
            {
                i++;
                Console.WriteLine(
                    $"{i}: {bucketObject.Key} \n\tBucket:
{bucketObject.BucketName}\n\tVersion: {bucketObject.VersionId}");
            }
        }

        return allObjects;
    }

    /// <summary>
    /// Present the user with the demo action choices.
    /// </summary>
    /// <returns>Async task.</returns>
    public static async Task<bool> DemoActionChoices()
```

```
{
    var choices = new string[]{
        "List all files in buckets.",
        "Attempt to delete a file.",
        "Attempt to delete a file with retention period bypass.",
        "Attempt to overwrite a file.",
        "View the object and bucket retention settings for a file.",
        "View the legal hold settings for a file.",
        "Finish the workflow."};

    var choice = 0;
    // Keep asking the user until they choose to move on.
    while (choice != 6)
    {
        Console.WriteLine(new string('-', 80));
        choice = GetChoiceResponse(
            "\nExplore the S3 locking features by selecting one of the
following choices:"
            , choices);
        Console.WriteLine(new string('-', 80));
        switch (choice)
        {
            case 0:
            {
                await ListBucketsAndObjects(true);
                break;
            }
            case 1:
            {
                Console.WriteLine("\nEnter the number of the object to
delete:");

                var allFiles = await ListBucketsAndObjects(true);
                var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
                await
_s3ActionsWrapper.DeleteObjectFromBucket(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, false, allFiles[fileChoice].VersionId);
                break;
            }
            case 2:
            {
                Console.WriteLine("\nEnter the number of the object to
delete:");

                var allFiles = await ListBucketsAndObjects(true);
```

```
        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        await
_s3ActionsWrapper.DeleteObjectFromBucket(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, true, allFiles[fileChoice].VersionId);
        break;
    }
    case 3:
    {
        var allFiles = await ListBucketsAndObjects(true);
        Console.WriteLine("\nEnter the number of the object to
overwrite:");

        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        // Create the file if it does not already exist.
        if (!File.Exists(allFiles[fileChoice].Key))
        {
            await using StreamWriter sw =
File.CreateText(allFiles[fileChoice].Key);
            await sw.WriteLineAsync(
                "This is a sample file for uploading to a
bucket.");
        }
        await
_s3ActionsWrapper.UploadFileAsync(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, allFiles[fileChoice].Key);
        break;
    }
    case 4:
    {
        var allFiles = await ListBucketsAndObjects(true);
        Console.WriteLine("\nEnter the number of the object and
bucket to view:");

        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        await
_s3ActionsWrapper.GetObjectRetention(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key);
        await
_s3ActionsWrapper.GetBucketObjectLockConfiguration(allFiles[fileChoice].BucketName);
        break;
    }
    case 5:
    {
```



```
        var allFiles = await ListBucketsAndObjects(true);
        Console.WriteLine("\nEnter the number of the object to
view:");

        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
        await
_s3ActionsWrapper.GetObjectLegalHold(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key);
        break;
    }
}
return true;
}

// <summary>
/// Clean up the resources from the scenario.
/// </summary>
/// <param name="interactive">True to run as interactive.</param>
/// <returns>True if successful.</returns>
public static async Task<bool> Cleanup(bool interactive)
{
    Console.WriteLine(new string('-', 80));

    if (!interactive || GetYesNoResponse("Do you want to clean up all files
and buckets? (y/n) "))
    {
        // Remove all locks and delete all buckets and objects.
        var allFiles = await ListBucketsAndObjects(false);
        foreach (var fileInfo in allFiles)
        {
            // Check for a legal hold.
            var legalHold = await
_s3ActionsWrapper.GetObjectLegalHold(fileInfo.BucketName, fileInfo.Key);
            if (legalHold?.Status?.Value == ObjectLockLegalHoldStatus.On)
            {
                await
_s3ActionsWrapper.ModifyObjectLegalHold(fileInfo.BucketName, fileInfo.Key,
ObjectLockLegalHoldStatus.Off);
            }

            // Check for a retention period.
            var retention = await
_s3ActionsWrapper.GetObjectRetention(fileInfo.BucketName, fileInfo.Key);
```

```

        var hasRetentionPeriod = retention?.Mode ==
ObjectLockRetentionMode.Governance && retention.RetainUntilDate >
DateTime.UtcNow.Date;
        await
_s3ActionsWrapper.DeleteObjectFromBucket(fileInfo.BucketName, fileInfo.Key,
hasRetentionPeriod, fileInfo.VersionId);
    }

    foreach (var bucketName in bucketNames)
    {
        await _s3ActionsWrapper.DeleteBucketByName(bucketName);
    }

}
else
{
    Console.WriteLine(
        "Ok, we'll leave the resources intact.\n" +
        "Don't forget to delete them when you're done with them or you
might incur unexpected charges."
    );
}

Console.WriteLine(new string('-', 80));
return true;
}

/// <summary>
/// Helper method to get a yes or no response from the user.
/// </summary>
/// <param name="question">The question string to print on the console.</
param>
/// <returns>True if the user responds with a yes.</returns>
private static bool GetYesNoResponse(string question)
{
    Console.WriteLine(question);
    var ynResponse = Console.ReadLine();
    var response = ynResponse != null && ynResponse.Equals("y",
StringComparison.InvariantCultureIgnoreCase);
    return response;
}

/// <summary>
/// Helper method to get a choice response from the user.

```

```
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <param name="choices">The choices to print on the console.</param>
    /// <returns>The index of the selected choice</returns>
    private static int GetChoiceResponse(string? question, string[] choices)
    {
        if (question != null)
        {
            Console.WriteLine(question);

            for (int i = 0; i < choices.Length; i++)
            {
                Console.WriteLine($"{i + 1}. {choices[i]}");
            }
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > choices.Length)
        {
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        return choiceNumber - 1;
    }
}
```

Eine Wrapper-Klasse für S3-Funktionen.

```
using System.Net;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

namespace S3ObjectLockScenario;

/// <summary>
/// Encapsulate the Amazon S3 operations.
/// </summary>
public class S3ActionsWrapper
```

```
{
    private readonly IAmazonS3 _amazonS3;

    /// <summary>
    /// Constructor for the S3ActionsWrapper.
    /// </summary>
    /// <param name="amazonS3">The injected S3 client.</param>
    public S3ActionsWrapper(IAmazonS3 amazonS3, IConfiguration configuration)
    {
        _amazonS3 = amazonS3;
    }

    /// <summary>
    /// Create a new Amazon S3 bucket with object lock actions.
    /// </summary>
    /// <param name="bucketName">The name of the bucket to create.</param>
    /// <param name="enableObjectLock">True to enable object lock on the
    bucket.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateBucketWithObjectLock(string bucketName, bool
    enableObjectLock)
    {
        Console.WriteLine($"\\tCreating bucket {bucketName} with object lock
    {enableObjectLock}.");
        try
        {
            var request = new PutBucketRequest
            {
                BucketName = bucketName,
                UseClientRegion = true,
                ObjectLockEnabledForBucket = enableObjectLock,
            };

            var response = await _amazonS3.PutBucketAsync(request);

            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error creating bucket: '{ex.Message}'");
            return false;
        }
    }
}
```

```
/// <summary>
/// Enable object lock on an existing bucket.
/// </summary>
/// <param name="bucketName">The name of the bucket to modify.</param>
/// <returns>True if successful.</returns>
public async Task<bool> EnableObjectLockOnBucket(string bucketName)
{
    try
    {
        // First, enable Versioning on the bucket.
        await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
        {
            BucketName = bucketName,
            VersioningConfig = new S3BucketVersioningConfig()
            {
                EnableMfaDelete = false,
                Status = VersionStatus.Enabled
            }
        });

        var request = new PutObjectLockConfigurationRequest()
        {
            BucketName = bucketName,
            ObjectLockConfiguration = new ObjectLockConfiguration()
            {
                ObjectLockEnabled = new ObjectLockEnabled("Enabled"),
            },
        };

        var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
        Console.WriteLine($" \tAdded an object lock policy to bucket
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error modifying object lock: '{ex.Message}'");
        return false;
    }
}

/// <summary>
```

```
/// Set or modify a retention period on an object in an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The key of the object.</param>
/// <param name="retention">The retention mode.</param>
/// <param name="retainUntilDate">The date retention expires.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectRetentionPeriod(string bucketName,
    string objectKey, ObjectLockRetentionMode retention, DateTime
retainUntilDate)
{
    try
    {
        var request = new PutObjectRetentionRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            Retention = new ObjectLockRetention()
            {
                Mode = retention,
                RetainUntilDate = retainUntilDate
            }
        };

        var response = await _amazonS3.PutObjectRetentionAsync(request);
        Console.WriteLine($"\\tSet retention for {objectKey} in {bucketName}
until {retainUntilDate:d}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tError modifying retention period:
'{ex.Message}'");
        return false;
    }
}

/// <summary>
/// Set or modify a retention period on an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket to modify.</param>
/// <param name="retention">The retention mode.</param>
/// <param name="retainUntilDate">The date for retention until.</param>
/// <returns>True if successful.</returns>
```

```
public async Task<bool> ModifyBucketDefaultRetention(string bucketName, bool
enableObjectLock, ObjectLockRetentionMode retention, DateTime retainUntilDate)
{
    var enabledString = enableObjectLock ? "Enabled" : "Disabled";
    var timeDifference = retainUntilDate.Subtract(DateTime.Now);
    try
    {
        // First, enable Versioning on the bucket.
        await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
        {
            BucketName = bucketName,
            VersioningConfig = new S3BucketVersioningConfig()
            {
                EnableMfaDelete = false,
                Status = VersionStatus.Enabled
            }
        });

        var request = new PutObjectLockConfigurationRequest()
        {
            BucketName = bucketName,
            ObjectLockConfiguration = new ObjectLockConfiguration()
            {
                ObjectLockEnabled = new ObjectLockEnabled(enabledString),
                Rule = new ObjectLockRule()
                {
                    DefaultRetention = new DefaultRetention()
                    {
                        Mode = retention,
                        Days = timeDifference.Days // Can be specified in
days or years but not both.
                    }
                }
            }
        };

        var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
        Console.WriteLine($"{bucketName}\tAdded a default retention to bucket
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
```

```
        {
            Console.WriteLine($"\\tError modifying object lock: '{ex.Message}'");
            return false;
        }
    }

    /// <summary>
    /// Get the retention period for an S3 object.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The object key.</param>
    /// <returns>The object retention details.</returns>
    public async Task<ObjectLockRetention> GetObjectRetention(string bucketName,
        string objectKey)
    {
        try
        {
            var request = new GetObjectRetentionRequest()
            {
                BucketName = bucketName,
                Key = objectKey
            };

            var response = await _amazonS3.GetObjectRetentionAsync(request);
            Console.WriteLine($"\\tObject retention for {objectKey} in
{bucketName}: " +
                $"\\n\\t{response.Retention.Mode} until
{response.Retention.RetainUntilDate:d}.");
            return response.Retention;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\\tUnable to fetch object lock retention:
'{ex.Message}'");
            return new ObjectLockRetention();
        }
    }

    /// <summary>
    /// Set or modify a legal hold on an object in an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The key of the object.</param>
    /// <param name="holdStatus">The On or Off status for the legal hold.</param>
```



```
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectLegalHold(string bucketName,
    string objectKey, ObjectLockLegalHoldStatus holdStatus)
{
    try
    {
        var request = new PutObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            LegalHold = new ObjectLockLegalHold()
            {
                Status = holdStatus
            }
        };

        var response = await _amazonS3.PutObjectLegalHoldAsync(request);
        Console.WriteLine($"\\tModified legal hold for {objectKey} in
{bucketName}.");
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tError modifying legal hold: '{ex.Message}'");
        return false;
    }
}

/// <summary>
/// Get the legal hold details for an S3 object.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object legal hold details.</returns>
public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };
    }
}
```

```
        var response = await _amazonS3.GetObjectLegalHoldAsync(request);
        Console.WriteLine($"\\tObject legal hold for {objectKey} in
{bucketName}: " +
                        $"\\n\\tStatus: {response.LegalHold.Status}");
        return response.LegalHold;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tUnable to fetch legal hold: '{ex.Message}'");
        return new ObjectLockLegalHold();
    }
}

/// <summary>
/// Get the object lock configuration details for an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket to get details.</param>
/// <returns>The bucket's object lock configuration details.</returns>
public async Task<ObjectLockConfiguration>
GetBucketObjectLockConfiguration(string bucketName)
{
    try
    {
        var request = new GetObjectLockConfigurationRequest()
        {
            BucketName = bucketName
        };

        var response = await
_amazonS3.GetObjectLockConfigurationAsync(request);
        Console.WriteLine($"\\tBucket object lock config for {bucketName} in
{bucketName}: " +
                        $"\\n\\tEnabled:
{response.ObjectLockConfiguration.ObjectLockEnabled}" +
                        $"\\n\\tRule:
{response.ObjectLockConfiguration.Rule?.DefaultRetention}");

        return response.ObjectLockConfiguration;
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"\\tUnable to fetch object lock config:
'{ex.Message}'");
    }
}
```

```
        return new ObjectLockConfiguration();
    }
}

/// <summary>
/// Upload a file from the local computer to an Amazon S3 bucket.
/// </summary>
/// <param name="bucketName">The Amazon S3 bucket to use.</param>
/// <param name="objectName">The object to upload.</param>
/// <param name="filePath">The path, including file name, of the object to
upload.</param>
/// <returns>True if success.</returns>
public async Task<bool> UploadFileAsync(string bucketName, string objectName,
string filePath)
{
    var request = new PutObjectRequest
    {
        BucketName = bucketName,
        Key = objectName,
        FilePath = filePath,
        ChecksumAlgorithm = ChecksumAlgorithm.SHA256
    };

    var response = await _amazonS3.PutObjectAsync(request);
    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
    {
        Console.WriteLine($"{\tSuccessfully uploaded {objectName} to
{bucketName}.");
        return true;
    }
    else
    {
        Console.WriteLine($"{\tCould not upload {objectName} to
{bucketName}.");
        return false;
    }
}

/// <summary>
/// List bucket objects and versions.
/// </summary>
/// <param name="bucketName">The Amazon S3 bucket to use.</param>
/// <returns>The list of objects and versions.</returns>
```

```
public async Task<ListVersionsResponse> ListBucketObjectsAndVersions(string
bucketName)
{
    var request = new ListVersionsRequest()
    {
        BucketName = bucketName
    };

    var response = await _amazonS3.ListVersionsAsync(request);
    return response;
}

/// <summary>
/// Delete an object from a specific bucket.
/// </summary>
/// <param name="bucketName">The Amazon S3 bucket to use.</param>
/// <param name="objectKey">The key of the object to delete.</param>
/// <param name="hasRetention">True if the object has retention settings.</
param>
/// <param name="versionId">Optional versionId.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteObjectFromBucket(string bucketName, string
objectKey, bool hasRetention, string? versionId = null)
{
    try
    {
        var request = new DeleteObjectRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            VersionId = versionId,
        };
        if (hasRetention)
        {
            // Set the BypassGovernanceRetention header
            // if the file has retention settings.
            request.BypassGovernanceRetention = true;
        }
        await _amazonS3.DeleteObjectAsync(request);
        Console.WriteLine(
            $"Deleted {objectKey} in {bucketName}.");
        return true;
    }
    catch (AmazonS3Exception ex)
```

```
        {
            Console.WriteLine($"\\tUnable to delete object {objectKey} in bucket
{bucketName}: " + ex.Message);
            return false;
        }
    }


    /// <summary>
    /// Delete a specific bucket.
    /// </summary>
    /// <param name="bucketName">The Amazon S3 bucket to use.</param>
    /// <param name="objectKey">The key of the object to delete.</param>
    /// <param name="versionId">Optional versionId.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteBucketByName(string bucketName)
    {
        try
        {
            var request = new DeleteBucketRequest() { BucketName = bucketName, };
            var response = await _amazonS3.DeleteBucketAsync(request);
            Console.WriteLine($"\\tDelete for {bucketName} complete.");
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\\tUnable to delete bucket {bucketName}: " +
ex.Message);
            return false;
        }
    }
}
```

- APIEinzelheiten finden Sie in den folgenden Themen unter AWS SDK for .NET APIReferenz.
 - [GetObjectLegalHold](#)
 - [GetObjectLockConfiguration](#)
 - [GetObjectRetention](#)
 - [PutObjectLegalHold](#)

- [PutObjectLockConfiguration](#)
- [PutObjectRetention](#)

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Führen Sie ein interaktives Szenario aus, in dem die Objektsperreffunktionen von Amazon S3 demonstriert werden.

```
// ObjectLockScenario contains the steps to run the S3 Object Lock workflow.
type ObjectLockScenario struct {
    questioner demotools.IQuestioner
    resources  Resources
    s3Actions  *actions.S3Actions
    sdkConfig  aws.Config
}

// NewObjectLockScenario constructs a new ObjectLockScenario instance.
func NewObjectLockScenario(sdkConfig aws.Config, questioner
demotools.IQuestioner) ObjectLockScenario {
    scenario := ObjectLockScenario{
        questioner: questioner,
        resources:  Resources{},
        s3Actions:  &actions.S3Actions{S3Client: s3.NewFromConfig(sdkConfig)},
        sdkConfig:  sdkConfig,
    }
    scenario.s3Actions.S3Manager = manager.NewUploader(scenario.s3Actions.S3Client)
    scenario.resources.init(scenario.s3Actions, questioner)
    return scenario
}

type nameLocked struct {
    name string
}
```

```
    locked bool
}

var createInfo = []nameLocked{
    {"standard-bucket", false},
    {"lock-bucket", true},
    {"retention-bucket", false},
}

// CreateBuckets creates the S3 buckets required for the workflow.
func (scenario *ObjectLockScenario) CreateBuckets(ctx context.Context) {
    log.Println("Let's create some S3 buckets to use for this workflow.")
    success := false
    for !success {
        prefix := scenario.questioner.Ask(
            "This example creates three buckets. Enter a prefix to name your buckets
            (remember bucket names must be globally unique):")

        for _, info := range createInfo {
            bucketName, err := scenario.s3Actions.CreateBucketWithLock(ctx,
                fmt.Sprintf("%s.%s", prefix, info.name), scenario.sdkConfig.Region, info.locked)
            if err != nil {
                switch err.(type) {
                    case *types.BucketAlreadyExists, *types.BucketAlreadyOwnedByYou:
                        log.Printf("Couldn't create bucket %s.\n", bucketName)
                    default:
                        panic(err)
                }
                break
            }
            scenario.resources.demoBuckets[info.name] = &DemoBucket{
                name:      bucketName,
                objectKeys: []string{},
            }
            log.Printf("Created bucket %s.\n", bucketName)
        }

        if len(scenario.resources.demoBuckets) < len(createInfo) {
            scenario.resources.deleteBuckets(ctx)
        } else {
            success = true
        }
    }
}
```

```
log.Println("S3 buckets created.")
log.Println(strings.Repeat("-", 88))
}

// EnableLockOnBucket enables object locking on an existing bucket.
func (scenario *ObjectLockScenario) EnableLockOnBucket(ctx context.Context) {
    log.Println("\nA bucket can be configured to use object locking.")
    scenario.questioner.Ask("Press Enter to continue.")

    var err error
    bucket := scenario.resources.demoBuckets["retention-bucket"]
    err = scenario.s3Actions.EnableObjectLockOnBucket(ctx, bucket.name)
    if err != nil {
        switch err.(type) {
        case *types.NoSuchBucket:
            log.Printf("Couldn't enable object locking on bucket %s.\n", bucket.name)
        default:
            panic(err)
        }
    } else {
        log.Printf("Object locking enabled on bucket %s.", bucket.name)
    }

    log.Println(strings.Repeat("-", 88))
}

// SetDefaultRetentionPolicy sets a default retention governance policy on a
bucket.
func (scenario *ObjectLockScenario) SetDefaultRetentionPolicy(ctx
context.Context) {
    log.Println("\nA bucket can be configured to use object locking with a default
retention period.")

    bucket := scenario.resources.demoBuckets["retention-bucket"]
    retentionPeriod := scenario.questioner.AskInt("Enter the default retention
period in days: ")
    err := scenario.s3Actions.ModifyDefaultBucketRetention(ctx,
bucket.name, types.ObjectLockEnabledEnabled, int32(retentionPeriod),
types.ObjectLockRetentionModeGovernance)
    if err != nil {
        switch err.(type) {
        case *types.NoSuchBucket:
            log.Printf("Couldn't configure a default retention period on bucket %s.\n",
bucket.name)
        }
    }
}
```



```
    default:
        panic(err)
    }
} else {
    log.Printf("Default retention policy set on bucket %s with %d day retention
period.", bucket.name, retentionPeriod)
    bucket.retentionEnabled = true
}

log.Println(strings.Repeat("-", 88))
}

// UploadTestObjects uploads test objects to the S3 buckets.
func (scenario *ObjectLockScenario) UploadTestObjects(ctx context.Context) {
    log.Println("Uploading test objects to S3 buckets.")

    for _, info := range createInfo {
        bucket := scenario.resources.demoBuckets[info.name]
        for i := 0; i < 2; i++ {
            key, err := scenario.s3Actions.UploadObject(ctx, bucket.name,
fmt.Sprintf("example-%d", i),
            fmt.Sprintf("Example object content #%d in bucket %s.", i, bucket.name))
            if err != nil {
                switch err.(type) {
                    case *types.NoSuchBucket:
                        log.Printf("Couldn't upload %s to bucket %s.\n", key, bucket.name)
                    default:
                        panic(err)
                }
            } else {
                log.Printf("Uploaded %s to bucket %s.\n", key, bucket.name)
                bucket.objectKeys = append(bucket.objectKeys, key)
            }
        }
    }
}

scenario.questioner.Ask("Test objects uploaded. Press Enter to continue.")
log.Println(strings.Repeat("-", 88))
}

// SetObjectLockConfigurations sets object lock configurations on the test
objects.
func (scenario *ObjectLockScenario) SetObjectLockConfigurations(ctx
context.Context) {
```

```
log.Println("Now let's set object lock configurations on individual objects.")

buckets := []*DemoBucket{scenario.resources.demoBuckets["lock-bucket"],
scenario.resources.demoBuckets["retention-bucket"]}
for _, bucket := range buckets {
    for index, objKey := range bucket.objectKeys {
        switch index {
        case 0:
            if scenario.questioner.AskBool(fmt.Sprintf("\nDo you want to add a legal hold
to %s in %s (y/n)? ", objKey, bucket.name), "y") {
                err := scenario.s3Actions.PutObjectLegalHold(ctx, bucket.name, objKey, "",
types.ObjectLockLegalHoldStatusOn)
                if err != nil {
                    switch err.(type) {
                    case *types.NoSuchKey:
                        log.Printf("Couldn't set legal hold on %s.\n", objKey)
                    default:
                        panic(err)
                    }
                } else {
                    log.Printf("Legal hold set on %s.\n", objKey)
                }
            }
        case 1:
            q := fmt.Sprintf("\nDo you want to add a 1 day Governance retention period to
%s in %s?\n"+
"Reminder: Only a user with the s3:BypassGovernanceRetention permission is
able to delete this object\n"+
"or its bucket until the retention period has expired. (y/n) ", objKey,
bucket.name)
            if scenario.questioner.AskBool(q, "y") {
                err := scenario.s3Actions.PutObjectRetention(ctx, bucket.name, objKey,
types.ObjectLockRetentionModeGovernance, 1)
                if err != nil {
                    switch err.(type) {
                    case *types.NoSuchKey:
                        log.Printf("Couldn't set retention period on %s in %s.\n", objKey,
bucket.name)
                    default:
                        panic(err)
                    }
                } else {
                    log.Printf("Retention period set to 1 for %s.", objKey)
                    bucket.retentionEnabled = true
                }
            }
        }
    }
}
```

```
    }
  }
}
}
}
log.Println(strings.Repeat("-", 88))
}

const (
  ListAll = iota
  DeleteObject
  DeleteRetentionObject
  OverwriteObject
  ViewRetention
  ViewLegalHold
  Finish
)

// InteractWithObjects allows the user to interact with the objects and test the
// object lock configurations.
func (scenario *ObjectLockScenario) InteractWithObjects(ctx context.Context) {
  log.Println("Now you can interact with the objects to explore the object lock
  configurations.")
  interactiveChoices := []string{
    "List all objects and buckets.",
    "Attempt to delete an object.",
    "Attempt to delete an object with retention period bypass.",
    "Attempt to overwrite a file.",
    "View the retention settings for an object.",
    "View the legal hold settings for an object.",
    "Finish the workflow."}

  choice := ListAll
  for choice != Finish {
    objList := scenario.GetAllObjects(ctx)
    objChoices := scenario.makeObjectChoiceList(objList)
    choice = scenario.questioner.AskChoice("Choose an action from the menu:\n",
    interactiveChoices)
    switch choice {
    case ListAll:
      log.Println("The current objects in the example buckets are:")
      for _, objChoice := range objChoices {
        log.Println("\t", objChoice)
      }
    }
  }
}
```

```
case DeleteObject, DeleteRetentionObject:
    objChoice := scenario.questioner.AskChoice("Enter the number of the object to
delete:\n", objChoices)
    obj := objList[objChoice]
    deleted, err := scenario.s3Actions.DeleteObject(ctx, obj.bucket, obj.key,
obj.versionId, choice == DeleteRetentionObject)
    if err != nil {
        switch err.(type) {
            case *types.NoSuchKey:
                log.Println("Nothing to delete.")
            default:
                panic(err)
        }
    } else if deleted {
        log.Printf("Object %s deleted.\n", obj.key)
    }
case OverwriteObject:
    objChoice := scenario.questioner.AskChoice("Enter the number of the object to
overwrite:\n", objChoices)
    obj := objList[objChoice]
    _, err := scenario.s3Actions.UploadObject(ctx, obj.bucket, obj.key,
fmt.Sprintf("New content in object %s.", obj.key))
    if err != nil {
        switch err.(type) {
            case *types.NoSuchBucket:
                log.Println("Couldn't upload to nonexistent bucket.")
            default:
                panic(err)
        }
    } else {
        log.Printf("Uploaded new content to object %s.\n", obj.key)
    }
case ViewRetention:
    objChoice := scenario.questioner.AskChoice("Enter the number of the object to
view:\n", objChoices)
    obj := objList[objChoice]
    retention, err := scenario.s3Actions.GetObjectRetention(ctx, obj.bucket,
obj.key)
    if err != nil {
        switch err.(type) {
            case *types.NoSuchKey:
                log.Printf("Can't get retention configuration for %s.\n", obj.key)
            default:
                panic(err)
        }
    }
```

```

    }
    } else if retention != nil {
        log.Printf("Object %s has retention mode %s until %v.\n", obj.key,
retention.Mode, retention.RetainUntilDate)
    } else {
        log.Printf("Object %s does not have object retention configured.\n", obj.key)
    }
    case ViewLegalHold:
        objChoice := scenario.questioner.AskChoice("Enter the number of the object to
view:\n", objChoices)
        obj := objList[objChoice]
        legalHold, err := scenario.s3Actions.GetObjectLegalHold(ctx, obj.bucket,
obj.key, obj.versionId)
        if err != nil {
            switch err.(type) {
            case *types.NoSuchKey:
                log.Printf("Can't get legal hold configuration for %s.\n", obj.key)
            default:
                panic(err)
            }
        } else if legalHold != nil {
            log.Printf("Object %s has legal hold %v.", obj.key, *legalHold)
        } else {
            log.Printf("Object %s does not have legal hold configured.", obj.key)
        }
    case Finish:
        log.Println("Let's clean up.")
    }
    log.Println(strings.Repeat("-", 88))
}
}

type BucketKeyVersionId struct {
    bucket    string
    key       string
    versionId string
}

// GetAllObjects gets the object versions in the example S3 buckets and returns
them in a flattened list.
func (scenario *ObjectLockScenario) GetAllObjects(ctx context.Context)
[]BucketKeyVersionId {
    var objectList []BucketKeyVersionId
    for _, info := range createInfo {

```

```

bucket := scenario.resources.demoBuckets[info.name]
versions, err := scenario.s3Actions.ListObjectVersions(ctx, bucket.name)
if err != nil {
    switch err.(type) {
    case *types.NoSuchBucket:
        log.Printf("Couldn't get object versions for %s.\n", bucket.name)
    default:
        panic(err)
    }
} else {
    for _, version := range versions {
        objectList = append(objectList,
            BucketKeyVersionId{bucket: bucket.name, key: *version.Key, versionId:
                *version.VersionId})
    }
}
return objectList
}

// makeObjectChoiceList makes the object version list into a list of strings that
// are displayed
// as choices.
func (scenario *ObjectLockScenario) makeObjectChoiceList(bucketObjects
    []BucketKeyVersionId) []string {
    choices := make([]string, len(bucketObjects))
    for i := 0; i < len(bucketObjects); i++ {
        choices[i] = fmt.Sprintf("%s in %s with VersionId %s.",
            bucketObjects[i].key, bucketObjects[i].bucket, bucketObjects[i].versionId)
    }
    return choices
}

// Run runs the S3 Object Lock workflow scenario.
func (scenario *ObjectLockScenario) Run(ctx context.Context) {
    defer func() {
        if r := recover(); r != nil {
            log.Println("Something went wrong with the demo.")
            _, isMock := scenario.questioner.(*demotools.MockQuestioner)
            if isMock || scenario.questioner.AskBool("Do you want to see the full error
                message (y/n)?", "y") {
                log.Println(r)
            }
        }
        scenario.resources.Cleanup(ctx)
    }
}

```

```

    }
  }()

  log.Println(strings.Repeat("-", 88))
  log.Println("Welcome to the Amazon S3 Object Lock Workflow Scenario.")
  log.Println(strings.Repeat("-", 88))

  scenario.CreateBuckets(ctx)
  scenario.EnableLockOnBucket(ctx)
  scenario.SetDefaultRetentionPolicy(ctx)
  scenario.UploadTestObjects(ctx)
  scenario.SetObjectLockConfigurations(ctx)
  scenario.InteractWithObjects(ctx)

  scenario.resources.Cleanup(ctx)

  log.Println(strings.Repeat("-", 88))
  log.Println("Thanks for watching!")
  log.Println(strings.Repeat("-", 88))
}

```

Definieren Sie eine Struktur, die die in diesem Beispiel verwendeten S3-Aktionen umschließt.

```

// S3Actions wraps S3 service actions.
type S3Actions struct {
  S3Client *s3.Client
  S3Manager *manager.Uploader
}

// CreateBucketWithLock creates a new S3 bucket with optional object locking
// enabled
// and waits for the bucket to exist before returning.
func (actor S3Actions) CreateBucketWithLock(ctx context.Context, bucket string,
  region string, enableObjectLock bool) (string, error) {
  input := &s3.CreateBucketInput{
    Bucket: aws.String(bucket),
    CreateBucketConfiguration: &types.CreateBucketConfiguration{
      LocationConstraint: types.BucketLocationConstraint(region),

```

```
    },
  }

  if enableObjectLock {
    input.ObjectLockEnabledForBucket = aws.Bool(true)
  }

  _, err := actor.S3Client.CreateBucket(ctx, input)
  if err != nil {
    var owned *types.BucketAlreadyOwnedByYou
    var exists *types.BucketAlreadyExists
    if errors.As(err, &owned) {
      log.Printf("You already own bucket %s.\n", bucket)
      err = owned
    } else if errors.As(err, &exists) {
      log.Printf("Bucket %s already exists.\n", bucket)
      err = exists
    }
  } else {
    err = s3.NewBucketExistsWaiter(actor.S3Client).Wait(
      ctx, &s3.HeadBucketInput{Bucket: aws.String(bucket)}, time.Minute)
    if err != nil {
      log.Printf("Failed attempt to wait for bucket %s to exist.\n", bucket)
    }
  }

  return bucket, err
}

// GetObjectLegalHold retrieves the legal hold status for an S3 object.
func (actor S3Actions) GetObjectLegalHold(ctx context.Context, bucket string, key
string, versionId string) (*types.ObjectLockLegalHoldStatus, error) {
  var status *types.ObjectLockLegalHoldStatus
  input := &s3.GetObjectLegalHoldInput{
    Bucket:    aws.String(bucket),
    Key:       aws.String(key),
    VersionId: aws.String(versionId),
  }

  output, err := actor.S3Client.GetObjectLegalHold(ctx, input)
  if err != nil {
    var noSuchKeyErr *types.NoSuchKey

```



```
var apiErr *smithy.GenericAPIError
if errors.As(err, &noSuchKeyErr) {
    log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
    err = noSuchKeyErr
} else if errors.As(err, &apiErr) {
    switch apiErr.ErrorCode() {
    case "NoSuchObjectLockConfiguration":
        log.Printf("Object %s does not have an object lock configuration.\n", key)
        err = nil
    case "InvalidRequest":
        log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
        err = nil
    }
}
} else {
    status = &output.LegalHold.Status
}

return status, err
}

// GetObjectLockConfiguration retrieves the object lock configuration for an S3
// bucket.
func (actor S3Actions) GetObjectLockConfiguration(ctx context.Context, bucket
string) (*types.ObjectLockConfiguration, error) {
    var lockConfig *types.ObjectLockConfiguration
    input := &s3.GetObjectLockConfigurationInput{
        Bucket: aws.String(bucket),
    }

    output, err := actor.S3Client.GetObjectLockConfiguration(ctx, input)
    if err != nil {
        var noBucket *types.NoSuchBucket
        var apiErr *smithy.GenericAPIError
        if errors.As(err, &noBucket) {
            log.Printf("Bucket %s does not exist.\n", bucket)
            err = noBucket
        } else if errors.As(err, &apiErr) && apiErr.ErrorCode() ==
"ObjectLockConfigurationNotFoundError" {
            log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
            err = nil
        }
    }
}
```

```
} else {
    lockConfig = output.ObjectLockConfiguration
}

return lockConfig, err
}

// GetObjectRetention retrieves the object retention configuration for an S3
// object.
func (actor S3Actions) GetObjectRetention(ctx context.Context, bucket string, key
string) (*types.ObjectLockRetention, error) {
    var retention *types.ObjectLockRetention
    input := &s3.GetObjectRetentionInput{
        Bucket: aws.String(bucket),
        Key:     aws.String(key),
    }

    output, err := actor.S3Client.GetObjectRetention(ctx, input)
    if err != nil {
        var noKey *types.NoSuchKey
        var apiErr *smithy.GenericAPIError
        if errors.As(err, &noKey) {
            log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
            err = noKey
        } else if errors.As(err, &apiErr) {
            switch apiErr.ErrorCode() {
            case "NoSuchObjectLockConfiguration":
                err = nil
            case "InvalidRequest":
                log.Printf("Bucket %s does not have locking enabled.", bucket)
                err = nil
            }
        }
    } else {
        retention = output.Retention
    }

    return retention, err
}
```

```
// PutObjectLegalHold sets the legal hold configuration for an S3 object.
func (actor S3Actions) PutObjectLegalHold(ctx context.Context, bucket string, key
string, versionId string, legalHoldStatus types.ObjectLockLegalHoldStatus) error
{
    input := &s3.PutObjectLegalHoldInput{
        Bucket: aws.String(bucket),
        Key:     aws.String(key),
        LegalHold: &types.ObjectLockLegalHold{
            Status: legalHoldStatus,
        },
    }
}
if versionId != "" {
    input.VersionId = aws.String(versionId)
}

_, err := actor.S3Client.PutObjectLegalHold(ctx, input)
if err != nil {
    var noKey *types.NoSuchKey
    if errors.As(err, &noKey) {
        log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
        err = noKey
    }
}

return err
}

// ModifyDefaultBucketRetention modifies the default retention period of an
existing bucket.
func (actor S3Actions) ModifyDefaultBucketRetention(
    ctx context.Context, bucket string, lockMode types.ObjectLockEnabled,
    retentionPeriod int32, retentionMode types.ObjectLockRetentionMode) error {

    input := &s3.PutObjectLockConfigurationInput{
        Bucket: aws.String(bucket),
        ObjectLockConfiguration: &types.ObjectLockConfiguration{
            ObjectLockEnabled: lockMode,
            Rule: &types.ObjectLockRule{
                DefaultRetention: &types.DefaultRetention{
                    Days: aws.Int32(retentionPeriod),
                    Mode: retentionMode,
                },
            },
        },
    }
}
```

```
    },
  },
}
_, err := actor.S3Client.PutObjectLockConfiguration(ctx, input)
if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
    log.Printf("Bucket %s does not exist.\n", bucket)
    err = noBucket
  }
}

return err
}

// EnableObjectLockOnBucket enables object locking on an existing bucket.
func (actor S3Actions) EnableObjectLockOnBucket(ctx context.Context, bucket
string) error {
  // Versioning must be enabled on the bucket before object locking is enabled.
  verInput := &s3.PutBucketVersioningInput{
    Bucket: aws.String(bucket),
    VersioningConfiguration: &types.VersioningConfiguration{
      MFADelete: types.MFADeleteDisabled,
      Status:    types.BucketVersioningStatusEnabled,
    },
  }
  _, err := actor.S3Client.PutBucketVersioning(ctx, verInput)
  if err != nil {
    var noBucket *types.NoSuchBucket
    if errors.As(err, &noBucket) {
      log.Printf("Bucket %s does not exist.\n", bucket)
      err = noBucket
    }
    return err
  }

  input := &s3.PutObjectLockConfigurationInput{
    Bucket: aws.String(bucket),
    ObjectLockConfiguration: &types.ObjectLockConfiguration{
      ObjectLockEnabled: types.ObjectLockEnabledEnabled,
    },
  }
}
```

```
_, err = actor.S3Client.PutObjectLockConfiguration(ctx, input)
if err != nil {
    var noBucket *types.NoSuchBucket
    if errors.As(err, &noBucket) {
        log.Printf("Bucket %s does not exist.\n", bucket)
        err = noBucket
    }
}

return err
}

// PutObjectRetention sets the object retention configuration for an S3 object.
func (actor S3Actions) PutObjectRetention(ctx context.Context, bucket string, key
string, retentionMode types.ObjectLockRetentionMode, retentionPeriodDays int32)
error {
    input := &s3.PutObjectRetentionInput{
        Bucket: aws.String(bucket),
        Key:     aws.String(key),
        Retention: &types.ObjectLockRetention{
            Mode:          retentionMode,
            RetainUntilDate: aws.Time(time.Now().AddDate(0, 0, int(retentionPeriodDays))),
        },
        BypassGovernanceRetention: aws.Bool(true),
    }

    _, err := actor.S3Client.PutObjectRetention(ctx, input)
    if err != nil {
        var noKey *types.NoSuchKey
        if errors.As(err, &noKey) {
            log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
            err = noKey
        }
    }

    return err
}

// UploadObject uses the S3 upload manager to upload an object to a bucket.
```

```
func (actor S3Actions) UploadObject(ctx context.Context, bucket string, key
string, contents string) (string, error) {
    var outKey string
    input := &s3.PutObjectInput{
        Bucket:      aws.String(bucket),
        Key:         aws.String(key),
        Body:        bytes.NewReader([]byte(contents)),
        ChecksumAlgorithm: types.ChecksumAlgorithmSha256,
    }
    output, err := actor.S3Manager.Upload(ctx, input)
    if err != nil {
        var noBucket *types.NoSuchBucket
        if errors.As(err, &noBucket) {
            log.Printf("Bucket %s does not exist.\n", bucket)
            err = noBucket
        }
    } else {
        err := s3.NewObjectExistsWaiter(actor.S3Client).Wait(ctx, &s3.HeadObjectInput{
            Bucket: aws.String(bucket),
            Key:    aws.String(key),
        }, time.Minute)
        if err != nil {
            log.Printf("Failed attempt to wait for object %s to exist in %s.\n", key,
            bucket)
        } else {
            outKey = *output.Key
        }
    }
    return outKey, err
}

// ListObjectVersions lists all versions of all objects in a bucket.
func (actor S3Actions) ListObjectVersions(ctx context.Context, bucket string)
([]types.ObjectVersion, error) {
    var err error
    var output *s3.ListObjectVersionsOutput
    var versions []types.ObjectVersion
    input := &s3.ListObjectVersionsInput{Bucket: aws.String(bucket)}
    versionPaginator := s3.NewListObjectVersionsPaginator(actor.S3Client, input)
    for versionPaginator.HasMorePages() {
        output, err = versionPaginator.NextPage(ctx)
        if err != nil {
```

```
    var noBucket *types.NoSuchBucket
    if errors.As(err, &noBucket) {
        log.Printf("Bucket %s does not exist.\n", bucket)
        err = noBucket
    }
    break
} else {
    versions = append(versions, output.Versions...)
}
}
return versions, err
}

// DeleteObject deletes an object from a bucket.
func (actor S3Actions) DeleteObject(ctx context.Context, bucket string, key
string, versionId string, bypassGovernance bool) (bool, error) {
    deleted := false
    input := &s3.DeleteObjectInput{
        Bucket: aws.String(bucket),
        Key:     aws.String(key),
    }
    if versionId != "" {
        input.VersionId = aws.String(versionId)
    }
    if bypassGovernance {
        input.BypassGovernanceRetention = aws.Bool(true)
    }
    _, err := actor.S3Client.DeleteObject(ctx, input)
    if err != nil {
        var noKey *types.NoSuchKey
        var apiErr *smithy.GenericAPIError
        if errors.As(err, &noKey) {
            log.Printf("Object %s does not exist in %s.\n", key, bucket)
            err = noKey
        } else if errors.As(err, &apiErr) {
            switch apiErr.ErrorCode() {
            case "AccessDenied":
                log.Printf("Access denied: cannot delete object %s from %s.\n", key, bucket)
                err = nil
            case "InvalidArgument":
                if bypassGovernance {
```

```
    log.Printf("You cannot specify bypass governance on a bucket without lock
enabled.")
    err = nil
}
}
} else {
    deleted = true
}
return deleted, err
}

// DeleteObjects deletes a list of objects from a bucket.
func (actor S3Actions) DeleteObjects(ctx context.Context, bucket string, objects
[]types.ObjectIdentifier, bypassGovernance bool) error {
    if len(objects) == 0 {
        return nil
    }

    input := s3.DeleteObjectsInput{
        Bucket: aws.String(bucket),
        Delete: &types.Delete{
            Objects: objects,
            Quiet:   aws.Bool(true),
        },
    }
    if bypassGovernance {
        input.BypassGovernanceRetention = aws.Bool(true)
    }
    delOut, err := actor.S3Client.DeleteObjects(ctx, &input)
    if err != nil || len(delOut.Errors) > 0 {
        log.Printf("Error deleting objects from bucket %s.\n", bucket)
        if err != nil {
            var noBucket *types.NoSuchBucket
            if errors.As(err, &noBucket) {
                log.Printf("Bucket %s does not exist.\n", bucket)
                err = noBucket
            }
        }
        } else if len(delOut.Errors) > 0 {
            for _, outErr := range delOut.Errors {
                log.Printf("%s: %s\n", *outErr.Key, *outErr.Message)
            }
        }
    }
}
```



```
    err = fmt.Errorf("%s", *delOut.Errors[0].Message)
  }
}
return err
}
```

Ressourcen bereinigen.

```
// DemoBucket contains metadata for buckets used in this example.
type DemoBucket struct {
    name            string
    retentionEnabled bool
    objectKeys      []string
}

// Resources keeps track of AWS resources created during the ObjectLockScenario
// and handles
// cleanup when the scenario finishes.
type Resources struct {
    demoBuckets map[string]*DemoBucket

    s3Actions  *actions.S3Actions
    questioner demotools.IQuestioner
}

// init initializes objects in the Resources struct.
func (resources *Resources) init(s3Actions *actions.S3Actions, questioner
    demotools.IQuestioner) {
    resources.s3Actions = s3Actions
    resources.questioner = questioner
    resources.demoBuckets = map[string]*DemoBucket{}
}

// Cleanup deletes all AWS resources created during the ObjectLockScenario.
func (resources *Resources) Cleanup(ctx context.Context) {
    defer func() {
        if r := recover(); r != nil {
            log.Printf("Something went wrong during cleanup.\n%v\n", r)
        }
    }()
}
```

```
    log.Println("Use the AWS Management Console to remove any remaining resources  
" +  
    "that were created for this scenario.")  
    }  
    }()  
  
    wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS  
resources that were created "+  
    "during this demo (y/n)?", "y")  
    if !wantDelete {  
        log.Println("Be sure to remove resources when you're done with them to avoid  
unexpected charges!")  
        return  
    }  
  
    log.Println("Removing objects from S3 buckets and deleting buckets...")  
    resources.deleteBuckets(ctx)  
    //resources.deleteRetentionObjects(resources.retentionBucket,  
resources.retentionObjects)  
  
    log.Println("Cleanup complete.")  
    }  
  
    // deleteBuckets empties and then deletes all buckets created during the  
ObjectLockScenario.  
func (resources *Resources) deleteBuckets(ctx context.Context) {  
    for _, info := range createInfo {  
        bucket := resources.demoBuckets[info.name]  
        resources.deleteObjects(ctx, bucket)  
        _, err := resources.s3Actions.S3Client.DeleteBucket(ctx, &s3.DeleteBucketInput{  
            Bucket: aws.String(bucket.name),  
        })  
        if err != nil {  
            panic(err)  
        }  
    }  
    resources.demoBuckets = map[string]*DemoBucket{}    }  
  
    // deleteObjects deletes all objects in the specified bucket.  
func (resources *Resources) deleteObjects(ctx context.Context, bucket  
*DemoBucket) {  
    lockConfig, err := resources.s3Actions.GetObjectLockConfiguration(ctx,  
bucket.name)
```

```
if err != nil {
    panic(err)
}
versions, err := resources.s3Actions.ListObjectVersions(ctx, bucket.name)
if err != nil {
    switch err.(type) {
    case *types.NoSuchBucket:
        log.Printf("No objects to get from %s.\n", bucket.name)
    default:
        panic(err)
    }
}
delObjects := make([]types.ObjectIdentifier, len(versions))
for i, version := range versions {
    if lockConfig != nil && lockConfig.ObjectLockEnabled ==
types.ObjectLockEnabledEnabled {
        status, err := resources.s3Actions.GetObjectLegalHold(ctx, bucket.name,
*version.Key, *version.VersionId)
        if err != nil {
            switch err.(type) {
            case *types.NoSuchKey:
                log.Printf("Couldn't determine legal hold status for %s in %s.\n",
*version.Key, bucket.name)
            default:
                panic(err)
            }
        } else if status != nil && *status == types.ObjectLockLegalHoldStatusOn {
            err = resources.s3Actions.PutObjectLegalHold(ctx, bucket.name, *version.Key,
*version.VersionId, types.ObjectLockLegalHoldStatusOff)
            if err != nil {
                switch err.(type) {
                case *types.NoSuchKey:
                    log.Printf("Couldn't turn off legal hold for %s in %s.\n", *version.Key,
bucket.name)
                default:
                    panic(err)
                }
            }
        }
        delObjects[i] = types.ObjectIdentifier{Key: version.Key, VersionId:
version.VersionId}
    }
}
```

```
err = resources.s3Actions.DeleteObjects(ctx, bucket.name, delObjects,
bucket.retentionEnabled)
if err != nil {
    switch err.(type) {
    case *types.NoSuchBucket:
        log.Println("Nothing to delete.")
    default:
        panic(err)
    }
}
}
```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for Go API-Referenz](#).
 - [GetObjectLegalHold](#)
 - [GetObjectLockConfiguration](#)
 - [GetObjectRetention](#)
 - [PutObjectLegalHold](#)
 - [PutObjectLockConfiguration](#)
 - [PutObjectRetention](#)

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Führen Sie ein interaktives Szenario aus, in dem die Objektsperrefunktionen von Amazon S3 demonstriert werden.

```
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHold;
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import java.io.BufferedWriter;
import java.io.IOException;
```

```
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;

/*
Before running this Java V2 code example, set up your development
environment, including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/setup.html

This Java example performs the following tasks:
  1. Create test Amazon Simple Storage Service (S3) buckets with different lock
  policies.
  2. Upload sample objects to each bucket.
  3. Set some Legal Hold and Retention Periods on objects and buckets.
  4. Investigate lock policies by viewing settings or attempting to delete or
  overwrite objects.
  5. Clean up objects and buckets.
*/
public class S3ObjectLockWorkflow {

    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");
    static String bucketName;
    static S3LockActions s3LockActions;
    private static final List<String> bucketNames = new ArrayList<>();
    private static final List<String> fileNames = new ArrayList<>();

    public static void main(String[] args) {
        // Get the current date and time to ensure bucket name is unique.
        LocalDateTime currentTime = LocalDateTime.now();

        // Format the date and time as a string.
        DateTimeFormatter formatter =
        DateTimeFormatter.ofPattern("yyyyMMddHHmmss");
        String timeStamp = currentTime.format(formatter);

        s3LockActions = new S3LockActions();
        bucketName = "bucket"+timeStamp;
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon Simple Storage Service (S3)
Object Locking Workflow Scenario.");
        System.out.println("Press Enter to continue...");
        scanner.nextLine();
        configurationSetup();
        System.out.println(DASHES);

        System.out.println(DASHES);
        setup();
        System.out.println("Setup is complete. Press Enter to continue...");
        scanner.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Lets present the user with choices.");
        System.out.println("Press Enter to continue...");
        scanner.nextLine();
        demoActionChoices() ;
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Would you like to clean up the resources? (y/n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            cleanup();
            System.out.println("Clean up is complete.");
        }

        System.out.println("Press Enter to continue...");
        scanner.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Amazon S3 Object Locking Workflow is complete.");
        System.out.println(DASHES);
    }

    // Present the user with the demo action choices.
    public static void demoActionChoices() {
        String[] choices = {
            "List all files in buckets.",
            "Attempt to delete a file.",
```

```
        "Attempt to delete a file with retention period bypass.",
        "Attempt to overwrite a file.",
        "View the object and bucket retention settings for a file.",
        "View the legal hold settings for a file.",
        "Finish the workflow."
    };

    int choice = 0;
    while (true) {
        System.out.println(DASHES);
        choice = getChoiceResponse("Explore the S3 locking features by
selecting one of the following choices:", choices);
        System.out.println(DASHES);
        System.out.println("You selected "+choices[choice]);
        switch (choice) {
            case 0 -> {
                s3LockActions.listBucketsAndObjects(bucketNames, true);
            }

            case 1 -> {
                System.out.println("Enter the number of the object to
delete:");

                List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
                List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
                String[] fileKeysArray = fileKeys.toArray(new String[0]);
                int fileChoice = getChoiceResponse(null, fileKeysArray);
                String objectKey = fileKeys.get(fileChoice);
                String bucketName = allFiles.get(fileChoice).getBucketName();
                String version = allFiles.get(fileChoice).getVersion();
                s3LockActions.deleteObjectFromBucket(bucketName, objectKey,
false, version);
            }

            case 2 -> {
                System.out.println("Enter the number of the object to
delete:");

                List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
                List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
                String[] fileKeysArray = fileKeys.toArray(new String[0]);
                int fileChoice = getChoiceResponse(null, fileKeysArray);
```

```
        String objectKey = fileKeys.get(fileChoice);
        String bucketName = allFiles.get(fileChoice).getBucketName();
        String version = allFiles.get(fileChoice).getVersion();
        s3LockActions.deleteObjectFromBucket(bucketName, objectKey,
true, version);
    }

    case 3 -> {
        System.out.println("Enter the number of the object to
overwrite:");
        List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
        List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
        String[] fileKeysArray = fileKeys.toArray(new String[0]);
        int fileChoice = getChoiceResponse(null, fileKeysArray);
        String objectKey = fileKeys.get(fileChoice);
        String bucketName = allFiles.get(fileChoice).getBucketName();

        // Attempt to overwrite the file.
        try (BufferedWriter writer = new BufferedWriter(new
java.io.FileWriter(objectKey))) {
            writer.write("This is a modified text.");
        } catch (IOException e) {
            e.printStackTrace();
        }
        s3LockActions.uploadFile(bucketName, objectKey, objectKey);
    }

    case 4 -> {
        System.out.println("Enter the number of the object to
overwrite:");
        List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
        List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
        String[] fileKeysArray = fileKeys.toArray(new String[0]);
        int fileChoice = getChoiceResponse(null, fileKeysArray);
        String objectKey = fileKeys.get(fileChoice);
        String bucketName = allFiles.get(fileChoice).getBucketName();
        s3LockActions.getObjectRetention(bucketName, objectKey);
    }
}
```



```
        case 5 -> {
            System.out.println("Enter the number of the object to
view:");

            List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
            List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
            String[] fileKeysArray = fileKeys.toArray(new String[0]);
            int fileChoice = getChoiceResponse(null, fileKeysArray);
            String objectKey = fileKeys.get(fileChoice);
            String bucketName = allFiles.get(fileChoice).getBucketName();
            s3LockActions.getObjectLegalHold(bucketName, objectKey);
            s3LockActions.getBucketObjectLockConfiguration(bucketName);
        }

        case 6 -> {
            System.out.println("Exiting the workflow...");
            return;
        }

        default -> {
            System.out.println("Invalid choice. Please select again.");
        }
    }
}

// Clean up the resources from the scenario.
private static void cleanup() {
    List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, false);
    for (S3InfoObject fileInfo : allFiles) {
        String bucketName = fileInfo.getBucketName();
        String key = fileInfo.getKeyName();
        String version = fileInfo.getVersion();
        if (bucketName.contains("lock-enabled") ||
(bucketName.contains("retention-after-creation"))) {
            ObjectLockLegalHold legalHold =
s3LockActions.getObjectLegalHold(bucketName, key);
            if (legalHold != null) {
                String holdStatus = legalHold.status().name();
                System.out.println(holdStatus);
                if (holdStatus.compareTo("ON") == 0) {
```

```
        s3LockActions.modifyObjectLegalHold(bucketName, key,
false);
    }
}
// Check for a retention period.
ObjectLockRetention retention =
s3LockActions.getObjectRetention(bucketName, key);
boolean hasRetentionPeriod ;
hasRetentionPeriod = retention != null;
s3LockActions.deleteObjectFromBucket(bucketName,
key,hasRetentionPeriod, version);

} else {
    System.out.println(bucketName +" objects do not have a legal
lock");
    s3LockActions.deleteObjectFromBucket(bucketName, key,false,
version);
}
}

// Delete the buckets.
System.out.println("Delete "+bucketName);
for (String bucket : bucketNames){
    s3LockActions.deleteBucketByName(bucket);
}
}

private static void setup() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("""
        For this workflow, we will use the AWS SDK for Java to create
several S3
        buckets and files to demonstrate working with S3 locking
features.
        """);

    System.out.println("S3 buckets can be created either with or without
object lock enabled.");
    System.out.println("Press Enter to continue...");
    scanner.nextLine();

    // Create three S3 buckets.
    s3LockActions.createBucketWithLockOptions(false, bucketNames.get(0));
    s3LockActions.createBucketWithLockOptions(true, bucketNames.get(1));
```

```
s3LockActions.createBucketWithLockOptions(false, bucketNames.get(2));
System.out.println("Press Enter to continue.");
scanner.nextLine();

System.out.println("Bucket "+bucketNames.get(2) +" will be configured to
use object locking with a default retention period.");
s3LockActions.modifyBucketDefaultRetention(bucketNames.get(2));
System.out.println("Press Enter to continue.");
scanner.nextLine();

System.out.println("Object lock policies can also be added to existing
buckets. For this example, we will use "+bucketNames.get(1));
s3LockActions.enableObjectLockOnBucket(bucketNames.get(1));
System.out.println("Press Enter to continue.");
scanner.nextLine();

// Upload some files to the buckets.
System.out.println("Now let's add some test files:");
String fileName = "exampleFile.txt";
int fileCount = 2;
try (BufferedWriter writer = new BufferedWriter(new
java.io.FileWriter(fileName))) {
    writer.write("This is a sample file for uploading to a bucket.");

} catch (IOException e) {
    e.printStackTrace();
}

for (String bucketName : bucketNames){
    for (int i = 0; i < fileCount; i++) {
        // Get the file name without extension.
        String fileNameWithoutExtension =
java.nio.file.Paths.get(fileName).getFileName().toString();
        int extensionIndex = fileNameWithoutExtension.lastIndexOf('.');
        if (extensionIndex > 0) {
            fileNameWithoutExtension =
fileNameWithoutExtension.substring(0, extensionIndex);
        }

        // Create the numbered file names.
        String numberedFileName = fileNameWithoutExtension + i +
getFileExtension(fileName);
        fileNames.add(numberedFileName);
        s3LockActions.uploadFile(bucketName, numberedFileName, fileName);
    }
}
```

```

    }
}

String question = null;
System.out.print("Press Enter to continue...");
scanner.nextLine();
System.out.println("Now we can set some object lock policies on
individual files:");
for (String bucketName : bucketNames) {
    for (int i = 0; i < fileNames.size(); i++){

        // No modifications to the objects in the first bucket.
        if (!bucketName.equals(bucketNames.get(0))) {
            String exampleFileName = fileNames.get(i);
            switch (i) {
                case 0 -> {
                    question = "Would you like to add a legal hold to " +
exampleFileName + " in " + bucketName + " (y/n)?";
                    System.out.println(question);
                    String ans = scanner.nextLine().trim();
                    if (ans.equalsIgnoreCase("y")) {
                        System.out.println("**** You have selected to put
a legal hold " + exampleFileName);

                            // Set a legal hold.
                            s3LockActions.modifyObjectLegalHold(bucketName,
exampleFileName, true);
                                }
                            }
                        case 1 -> {
                            """"
                                Would you like to add a 1 day Governance
retention period to %s in %s (y/n)?
                                Reminder: Only a user with the
s3:BypassGovernanceRetention permission will be able to delete this file or its
bucket until the retention period has expired.
                                """".formatted(exampleFileName, bucketName);
                                    System.out.println(question);
                                    String ans2 = scanner.nextLine().trim();
                                    if (ans2.equalsIgnoreCase("y")) {

s3LockActions.modifyObjectRetentionPeriod(bucketName, exampleFileName);
                                        }
                                    }
                                }
                            }
}
}

```

```
        }
    }
}

// Get file extension.
private static String getFileExtension(String fileName) {
    int dotIndex = fileName.lastIndexOf('.');
    if (dotIndex > 0) {
        return fileName.substring(dotIndex);
    }
    return "";
}

public static void configurationSetup() {
    String noLockBucketName = bucketName + "-no-lock";
    String lockEnabledBucketName = bucketName + "-lock-enabled";
    String retentionAfterCreationBucketName = bucketName + "-retention-after-
creation";
    bucketNames.add(noLockBucketName);
    bucketNames.add(lockEnabledBucketName);
    bucketNames.add(retentionAfterCreationBucketName);
}

public static int getChoiceResponse(String question, String[] choices) {
    Scanner scanner = new Scanner(System.in);
    if (question != null) {
        System.out.println(question);
        for (int i = 0; i < choices.length; i++) {
            System.out.println("\t" + (i + 1) + ". " + choices[i]);
        }
    }

    int choiceNumber = 0;
    while (choiceNumber < 1 || choiceNumber > choices.length) {
        String choice = scanner.nextLine();
        try {
            choiceNumber = Integer.parseInt(choice);
        } catch (NumberFormatException e) {
            System.out.println("Invalid choice. Please enter a valid
number.");
        }
    }
}
```

```
        return choiceNumber - 1;
    }
}
```

Eine Wrapper-Klasse für S3-Funktionen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketVersioningStatus;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.CreateBucketRequest;
import software.amazon.awssdk.services.s3.model.DefaultRetention;
import software.amazon.awssdk.services.s3.model.DeleteBucketRequest;
import software.amazon.awssdk.services.s3.model.DeleteObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectLegalHoldRequest;
import software.amazon.awssdk.services.s3.model.GetObjectLegalHoldResponse;
import
    software.amazon.awssdk.services.s3.model.GetObjectLockConfigurationRequest;
import
    software.amazon.awssdk.services.s3.model.GetObjectLockConfigurationResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.GetObjectRetentionResponse;
import software.amazon.awssdk.services.s3.model.HeadBucketRequest;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsRequest;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsResponse;
import software.amazon.awssdk.services.s3.model.MFADelete;
import software.amazon.awssdk.services.s3.model.ObjectLockConfiguration;
import software.amazon.awssdk.services.s3.model.ObjectLockEnabled;
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHold;
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHoldStatus;
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import software.amazon.awssdk.services.s3.model.ObjectLockRetentionMode;
import software.amazon.awssdk.services.s3.model.ObjectLockRule;
import software.amazon.awssdk.services.s3.model.PutBucketVersioningRequest;
import software.amazon.awssdk.services.s3.model.PutObjectLegalHoldRequest;
import
    software.amazon.awssdk.services.s3.model.PutObjectLockConfigurationRequest;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.PutObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
```

```
import software.amazon.awssdk.services.s3.model.VersioningConfiguration;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.List;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.stream.Collectors;

// Contains application logic for the Amazon S3 operations used in this workflow.
public class S3LockActions {

    private static S3Client getClient() {
        return S3Client.builder()
            .region(Region.US_EAST_1)
            .build();
    }

    // Set or modify a retention period on an object in an S3 bucket.
    public void modifyObjectRetentionPeriod(String bucketName, String objectKey)
    {
        // Calculate the instant one day from now.
        Instant futureInstant = Instant.now().plus(1, ChronoUnit.DAYS);

        // Convert the Instant to a ZonedDateTime object with a specific time
        zone.
        ZonedDateTime zonedDateTime =
        futureInstant.atZone(ZoneId.systemDefault());

        // Define a formatter for human-readable output.
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
        HH:mm:ss");

        // Format the ZonedDateTime object to a human-readable date string.
        String humanReadableDate = formatter.format(zonedDateTime);

        // Print the formatted date string.
        System.out.println("Formatted Date: " + humanReadableDate);
        ObjectLockRetention retention = ObjectLockRetention.builder()
            .mode(ObjectLockRetentionMode.GOVERNANCE)
```

```
        .retainUntilDate(futureInstant)
        .build();

    PutObjectRetentionRequest retentionRequest =
PutObjectRetentionRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .retention(retention)
        .build();

    getClient().putObjectRetention(retentionRequest);
    System.out.println("Set retention for "+objectKey +" in " +bucketName +"
until "+ humanReadableDate +".");
}

// Get the legal hold details for an S3 object.
public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
    try {
        GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .build();

        GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
        System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
            ":\n\tStatus: " + response.legalHold().status());
        return response.legalHold();

    } catch (S3Exception ex) {
        System.out.println("\tUnable to fetch legal hold: '" +
ex.getMessage() + "'");
    }

    return null;
}

// Create a new Amazon S3 bucket with object lock options.
public void createBucketWithLockOptions(boolean enableObjectLock, String
bucketName) {
    S3Waiter s3Waiter = getClient().waiter();
```



```
        CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
            .bucket(bucketName)
            .objectLockEnabledForBucket(enableObjectLock)
            .build();

        getClient().createBucket(bucketRequest);
        HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
            .bucket(bucketName)
            .build();

        // Wait until the bucket is created and print out the response.
        s3Waiter.waitUntilBucketExists(bucketRequestWait);
        System.out.println(bucketName + " is ready");
    }

    public List<S3InfoObject> listBucketsAndObjects(List<String> bucketNames,
        Boolean interactive) {
        AtomicInteger counter = new AtomicInteger(0); // Initialize counter.
        return bucketNames.stream()
            .flatMap(bucketName ->
                listBucketObjectsAndVersions(bucketName).versions().stream()
                    .map(version -> {
                        S3InfoObject s3InfoObject = new S3InfoObject();
                        s3InfoObject.setBucketName(bucketName);
                        s3InfoObject.setVersion(version.versionId());
                        s3InfoObject.setKeyName(version.key());
                        return s3InfoObject;
                    })))
            .peek(s3InfoObject -> {
                int i = counter.incrementAndGet(); // Increment and get the
                updated value.
                if (interactive) {
                    System.out.println(i + ": " + s3InfoObject.getKeyName());
                    System.out.printf("%5s Bucket name: %s\n", "",
                        s3InfoObject.getBucketName());
                    System.out.printf("%5s Version: %s\n", "",
                        s3InfoObject.getVersion());
                }
            })
            .collect(Collectors.toList());
    }

    public ListObjectVersionsResponse listBucketObjectsAndVersions(String
        bucketName) {
```

```
ListObjectVersionsRequest versionsRequest =
ListObjectVersionsRequest.builder()
    .bucket(bucketName)
    .build();

return getClient().listObjectVersions(versionsRequest);
}

// Set or modify a retention period on an S3 bucket.
public void modifyBucketDefaultRetention(String bucketName) {
    VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
    .mfaDelete(MFADelete.DISABLED)
    .status(BucketVersioningStatus.ENABLED)
    .build();

    PutBucketVersioningRequest versioningRequest =
PutBucketVersioningRequest.builder()
    .bucket(bucketName)
    .versioningConfiguration(versioningConfiguration)
    .build();

    getClient().putBucketVersioning(versioningRequest);
    DefaultRetention retention = DefaultRetention.builder()
    .days(1)
    .mode(ObjectLockRetentionMode.GOVERNANCE)
    .build();

    ObjectLockRule lockRule = ObjectLockRule.builder()
    .defaultRetention(retention)
    .build();

    ObjectLockConfiguration objectLockConfiguration =
ObjectLockConfiguration.builder()
    .objectLockEnabled(ObjectLockEnabled.ENABLED)
    .rule(lockRule)
    .build();

    PutObjectLockConfigurationRequest putObjectLockConfigurationRequest =
PutObjectLockConfigurationRequest.builder()
    .bucket(bucketName)
    .objectLockConfiguration(objectLockConfiguration)
    .build();
```

```
getClient().putObjectLockConfiguration(putObjectLockConfigurationRequest) ;
    System.out.println("Added a default retention to bucket "+bucketName
+ ".");
}

// Enable object lock on an existing bucket.
public void enableObjectLockOnBucket(String bucketName) {
    try {
        VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
            .status(BucketVersioningStatus.ENABLED)
            .build();

        PutBucketVersioningRequest putBucketVersioningRequest =
PutBucketVersioningRequest.builder()
            .bucket(bucketName)
            .versioningConfiguration(versioningConfiguration)
            .build();

        // Enable versioning on the bucket.
        getClient().putBucketVersioning(putBucketVersioningRequest);
        PutObjectLockConfigurationRequest request =
PutObjectLockConfigurationRequest.builder()
            .bucket(bucketName)
            .objectLockConfiguration(ObjectLockConfiguration.builder()
                .objectLockEnabled(ObjectLockEnabled.ENABLED)
                .build())
            .build();

        getClient().putObjectLockConfiguration(request);
        System.out.println("Successfully enabled object lock on
"+bucketName);

    } catch (S3Exception ex) {
        System.out.println("Error modifying object lock: '" + ex.getMessage()
+ "'");
    }
}

public void uploadFile(String bucketName, String objectName, String filePath)
{
    Path file = Paths.get(filePath);
    PutObjectRequest request = PutObjectRequest.builder()
```

```
        .bucket(bucketName)
        .key(objectName)
        .checksumAlgorithm(ChecksumAlgorithm.SHA256)
        .build();

    PutObjectResponse response = getClient().putObject(request, file);
    if (response != null) {
        System.out.println("\tSuccessfully uploaded " + objectName + " to " +
bucketName + ".");
    } else {
        System.out.println("\tCould not upload " + objectName + " to " +
bucketName + ".");
    }
}

// Set or modify a legal hold on an object in an S3 bucket.
public void modifyObjectLegalHold(String bucketName, String objectKey,
boolean legalHoldOn) {
    ObjectLockLegalHold legalHold ;
    if (legalHoldOn) {
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.ON)
            .build();
    } else {
        legalHold = ObjectLockLegalHold.builder()
            .status(ObjectLockLegalHoldStatus.OFF)
            .build();
    }

    PutObjectLegalHoldRequest legalHoldRequest =
PutObjectLegalHoldRequest.builder()
        .bucket(bucketName)
        .key(objectKey)
        .legalHold(legalHold)
        .build();

    getClient().putObjectLegalHold(legalHoldRequest) ;
    System.out.println("Modified legal hold for "+ objectKey +" in
"+"bucketName +".");
}

// Delete an object from a specific bucket.
public void deleteObjectFromBucket(String bucketName, String objectKey,
boolean hasRetention, String versionId) {
```

```
    try {
        DeleteObjectRequest objectRequest;
        if (hasRetention) {
            objectRequest = DeleteObjectRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .versionId(versionId)
                .bypassGovernanceRetention(true)
                .build();
        } else {
            objectRequest = DeleteObjectRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .versionId(versionId)
                .build();
        }

        getClient().deleteObject(objectRequest) ;
        System.out.println("The object was successfully deleted");

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
    }
}

// Get the retention period for an S3 object.
public ObjectLockRetention getObjectRetention(String bucketName, String key){
    try {
        GetObjectRetentionRequest retentionRequest =
GetObjectRetentionRequest.builder()
            .bucket(bucketName)
            .key(key)
            .build();

        GetObjectRetentionResponse response =
getClient().getObjectRetention(retentionRequest);
        System.out.println("tObject retention for "+key +"
in "+ bucketName +": " + response.retention().mode() +" until "+
response.retention().retainUntilDate() +".");
        return response.retention();

    } catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        return null;
    }
}
```

```
    }  
  }  
  
  public void deleteBucketByName(String bucketName) {  
    try {  
      DeleteBucketRequest request = DeleteBucketRequest.builder()  
        .bucket(bucketName)  
        .build();  
  
      getClient().deleteBucket(request);  
      System.out.println(bucketName + " was deleted.");  
  
    } catch (S3Exception e) {  
      System.err.println(e.awsErrorDetails().errorMessage());  
    }  
  }  
  
  // Get the object lock configuration details for an S3 bucket.  
  public void getBucketObjectLockConfiguration(String bucketName) {  
    GetObjectLockConfigurationRequest objectLockConfigurationRequest =  
    GetObjectLockConfigurationRequest.builder()  
      .bucket(bucketName)  
      .build();  
  
    GetObjectLockConfigurationResponse response =  
    getClient().getObjectLockConfiguration(objectLockConfigurationRequest);  
    System.out.println("Bucket object lock config for "+bucketName +": ");  
    System.out.println("\tEnabled:  
"+response.getObjectLockConfiguration().getObjectLockEnabled());  
    System.out.println("\tRule: "+  
    response.getObjectLockConfiguration().rule().defaultRetention());  
  }  
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK for Java 2.x API-Referenz.
 - [GetObjectLegalHold](#)
 - [GetObjectLockConfiguration](#)
 - [GetObjectRetention](#)
 - [PutObjectLegalHold](#)

- [PutObjectLockConfiguration](#)
- [PutObjectRetention](#)

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

index.js - Einstiegspunkt für den Workflow. Dadurch werden alle Schritte orchestriert. Die Implementierungsdetails für Szenario, ScenarioInput ScenarioOutput, und ScenarioAction finden GitHub Sie unter.

```
import * as Scenarios from "@aws-doc-sdk-examples/lib/scenario/index.js";
import {
  exitOnFalse,
  loadState,
  saveState,
} from "@aws-doc-sdk-examples/lib/scenario/steps-common.js";

import { welcome, welcomeContinue } from "./welcome.steps.js";
import {
  confirmCreateBuckets,
  confirmPopulateBuckets,
  confirmSetLegalHoldFileEnabled,
  confirmSetLegalHoldFileRetention,
  confirmSetRetentionPeriodFileEnabled,
  confirmSetRetentionPeriodFileRetention,
  confirmUpdateLockPolicy,
  confirmUpdateRetention,
  createBuckets,
  createBucketsAction,
  populateBuckets,
  populateBucketsAction,
  setLegalHoldFileEnabledAction,
  setLegalHoldFileRetentionAction,
  setRetentionPeriodFileEnabledAction,
```

```
    setRetentionPeriodFileRetentionAction,  
    updateLockPolicy,  
    updateLockPolicyAction,  
    updateRetention,  
    updateRetentionAction,  
  } from "./setup.steps.js";  
  
/**  
 * @param {Scenarios} scenarios  
 * @param {Record<string, any>} initialState  
 */  
export const getWorkflowStages = (scenarios, initialState = {}) => {  
  const client = new S3Client({});  
  
  return {  
    deploy: new scenarios.Scenario(  
      "S3 Object Locking - Deploy",  
      [  
        welcome(scenarios),  
        welcomeContinue(scenarios),  
        exitOnFalse(scenarios, "welcomeContinue"),  
        createBuckets(scenarios),  
        confirmCreateBuckets(scenarios),  
        exitOnFalse(scenarios, "confirmCreateBuckets"),  
        createBucketsAction(scenarios, client),  
        updateRetention(scenarios),  
        confirmUpdateRetention(scenarios),  
        exitOnFalse(scenarios, "confirmUpdateRetention"),  
        updateRetentionAction(scenarios, client),  
        populateBuckets(scenarios),  
        confirmPopulateBuckets(scenarios),  
        exitOnFalse(scenarios, "confirmPopulateBuckets"),  
        populateBucketsAction(scenarios, client),  
        updateLockPolicy(scenarios),  
        confirmUpdateLockPolicy(scenarios),  
        exitOnFalse(scenarios, "confirmUpdateLockPolicy"),  
        updateLockPolicyAction(scenarios, client),  
        confirmSetLegalHoldFileEnabled(scenarios),  
        setLegalHoldFileEnabledAction(scenarios, client),  
        confirmSetRetentionPeriodFileEnabled(scenarios),  
        setRetentionPeriodFileEnabledAction(scenarios, client),  
        confirmSetLegalHoldFileRetention(scenarios),  
        setLegalHoldFileRetentionAction(scenarios, client),  
        confirmSetRetentionPeriodFileRetention(scenarios),
```



```

        setRetentionPeriodFileRetentionAction(scenarios, client),
        saveState,
    ],
    initialState,
),
demo: new scenarios.Scenario(
    "S3 Object Locking - Demo",
    [loadState, replAction(scenarios, client)],
    initialState,
),
clean: new scenarios.Scenario(
    "S3 Object Locking - Destroy",
    [
        loadState,
        confirmCleanup(scenarios),
        exitOnFalse(scenarios, "confirmCleanup"),
        cleanupAction(scenarios, client),
    ],
    initialState,
),
};
};

// Call function if run directly
import { fileURLToPath } from "url";
import { S3Client } from "@aws-sdk/client-s3";
import { cleanupAction, confirmCleanup } from "./clean.steps.js";
import { replAction } from "./repl.steps.js";

if (process.argv[1] === fileURLToPath(import.meta.url)) {
    const objectLockingScenarios = getWorkflowStages(Scenarios);
    Scenarios.parseScenarioArgs(objectLockingScenarios);
}

```

welcome.steps.js - Gibt Willkommensnachrichten an die Konsole aus.

```

/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
 */

/**
 * @param {Scenarios} scenarios

```

```
*/
const welcome = (scenarios) =>
  new scenarios.ScenarioOutput(
    "welcome",
    `Welcome to the Amazon Simple Storage Service (S3) Object Locking Workflow
    Scenario. For this workflow, we will use the AWS SDK for JavaScript to create
    several S3 buckets and files to demonstrate working with S3 locking features.` ,
    { header: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const welcomeContinue = (scenarios) =>
  new scenarios.ScenarioInput(
    "welcomeContinue",
    "Press Enter when you are ready to start.",
    { type: "confirm" },
  );

export { welcome, welcomeContinue };
```

setup.steps.js - Stellen Sie Buckets, Objekte und Dateieinstellungen bereit.

```
import {
  BucketVersioningStatus,
  ChecksumAlgorithm,
  CreateBucketCommand,
  MFADeleteStatus,
  PutBucketVersioningCommand,
  PutObjectCommand,
  PutObjectLockConfigurationCommand,
  PutObjectLegalHoldCommand,
  PutObjectRetentionCommand,
  ObjectLockLegalHoldStatus,
  ObjectLockRetentionMode,
} from "@aws-sdk/client-s3";

/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
 */
```

```
/**
 * @typedef {import("@aws-sdk/client-s3").S3Client} S3Client
 */

const bucketPrefix = "js-object-locking";

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const createBuckets = (scenarios) =>
  new scenarios.ScenarioOutput(
    "createBuckets",
    `The following buckets will be created:
      ${bucketPrefix}-no-lock with object lock False.
      ${bucketPrefix}-lock-enabled with object lock True.
      ${bucketPrefix}-retention-after-creation with object lock False.`,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmCreateBuckets = (scenarios) =>
  new scenarios.ScenarioInput("confirmCreateBuckets", "Create the buckets?", {
    type: "confirm",
  });

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const createBucketsAction = (scenarios, client) =>
  new scenarios.ScenarioAction("createBucketsAction", async (state) => {
    const noLockBucketName = `${bucketPrefix}-no-lock`;
    const lockEnabledBucketName = `${bucketPrefix}-lock-enabled`;
    const retentionBucketName = `${bucketPrefix}-retention-after-creation`;

    await client.send(new CreateBucketCommand({ Bucket: noLockBucketName }));
    await client.send(
      new CreateBucketCommand({
        Bucket: lockEnabledBucketName,
        ObjectLockEnabledForBucket: true,
      }),
    ),
  ),
);
```

```
    );
    await client.send(new CreateBucketCommand({ Bucket: retentionBucketName }));

    state.noLockBucketName = noLockBucketName;
    state.lockEnabledBucketName = lockEnabledBucketName;
    state.retentionBucketName = retentionBucketName;
  });

/**
 * @param {Scenarios} scenarios
 */
const populateBuckets = (scenarios) =>
  new scenarios.ScenarioOutput(
    "populateBuckets",
    `The following test files will be created:
      file0.txt in ${bucketPrefix}-no-lock.
      file1.txt in ${bucketPrefix}-no-lock.
      file0.txt in ${bucketPrefix}-lock-enabled.
      file1.txt in ${bucketPrefix}-lock-enabled.
      file0.txt in ${bucketPrefix}-retention-after-creation.
      file1.txt in ${bucketPrefix}-retention-after-creation.` ,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmPopulateBuckets = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmPopulateBuckets",
    "Populate the buckets?",
    { type: "confirm" },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const populateBucketsAction = (scenarios, client) =>
  new scenarios.ScenarioAction("populateBucketsAction", async (state) => {
    await client.send(
      new PutObjectCommand({
        Bucket: state.noLockBucketName,
        Key: "file0.txt",
```

```
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
    })),
);
await client.send(
    new PutObjectCommand({
        Bucket: state.noLockBucketName,
        Key: "file1.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
    })),
);
await client.send(
    new PutObjectCommand({
        Bucket: state.lockEnabledBucketName,
        Key: "file0.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
    })),
);
await client.send(
    new PutObjectCommand({
        Bucket: state.lockEnabledBucketName,
        Key: "file1.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
    })),
);
await client.send(
    new PutObjectCommand({
        Bucket: state.retentionBucketName,
        Key: "file0.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
    })),
);
await client.send(
    new PutObjectCommand({
        Bucket: state.retentionBucketName,
        Key: "file1.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
    })),
);
```

```
});

/**
 * @param {Scenarios} scenarios
 */
const updateRetention = (scenarios) =>
  new scenarios.ScenarioOutput(
    "updateRetention",
    `A bucket can be configured to use object locking with a default retention
    period.
    A default retention period will be configured for ${bucketPrefix}-retention-
    after-creation.` ,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmUpdateRetention = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmUpdateRetention",
    "Configure default retention period?",
    { type: "confirm" },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const updateRetentionAction = (scenarios, client) =>
  new scenarios.ScenarioAction("updateRetentionAction", async (state) => {
    await client.send(
      new PutBucketVersioningCommand({
        Bucket: state.retentionBucketName,
        VersioningConfiguration: {
          MFADelete: MFADeleteStatus.Disabled,
          Status: BucketVersioningStatus.Enabled,
        },
      }),
    );

    await client.send(
      new PutObjectLockConfigurationCommand({
        Bucket: state.retentionBucketName,
```

```

        ObjectLockConfiguration: {
          ObjectLockEnabled: "Enabled",
          Rule: {
            DefaultRetention: {
              Mode: "GOVERNANCE",
              Years: 1,
            },
          },
        },
      })),
    );
  });

/**
 * @param {Scenarios} scenarios
 */
const updateLockPolicy = (scenarios) =>
  new scenarios.ScenarioOutput(
    "updateLockPolicy",
    `Object lock policies can also be added to existing buckets.
    An object lock policy will be added to ${bucketPrefix}-lock-enabled.` ,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmUpdateLockPolicy = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmUpdateLockPolicy",
    "Add object lock policy?",
    { type: "confirm" },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const updateLockPolicyAction = (scenarios, client) =>
  new scenarios.ScenarioAction("updateLockPolicyAction", async (state) => {
    await client.send(
      new PutObjectLockConfigurationCommand({
        Bucket: state.lockEnabledBucketName,
        ObjectLockConfiguration: {

```

```
        ObjectLockEnabled: "Enabled",
      },
    )),
  );
});

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const confirmSetLegalHoldFileEnabled = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetLegalHoldFileEnabled",
    (state) =>
      `Would you like to add a legal hold to file0.txt in
    ${state.lockEnabledBucketName}?`,
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setLegalHoldFileEnabledAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setLegalHoldFileEnabledAction",
    async (state) => {
      await client.send(
        new PutObjectLegalHoldCommand({
          Bucket: state.lockEnabledBucketName,
          Key: "file0.txt",
          LegalHold: {
            Status: ObjectLockLegalHoldStatus.ON,
          },
        }),
      );
      console.log(
        `Modified legal hold for file0.txt in ${state.lockEnabledBucketName}.`,
      );
    },
    { skipWhen: (state) => !state.confirmSetLegalHoldFileEnabled },
  );
```



```
/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const confirmSetRetentionPeriodFileEnabled = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetRetentionPeriodFileEnabled",
    (state) =>
      `Would you like to add a 1 day Governance retention period to file1.txt in
      ${state.lockEnabledBucketName}?
      Reminder: Only a user with the s3:BypassGovernanceRetention permission will be
      able to delete this file or its bucket until the retention period has expired.`
    ,
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setRetentionPeriodFileEnabledAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setRetentionPeriodFileEnabledAction",
    async (state) => {
      const retentionDate = new Date();
      retentionDate.setDate(retentionDate.getDate() + 1);
      await client.send(
        new PutObjectRetentionCommand({
          Bucket: state.lockEnabledBucketName,
          Key: "file1.txt",
          Retention: {
            Mode: ObjectLockRetentionMode.GOVERNANCE,
            RetainUntilDate: retentionDate,
          },
        })
      );
      console.log(
        `Set retention for file1.txt in ${state.lockEnabledBucketName} until
        ${retentionDate.toISOString().split("T")[0]}.`
      );
    },
    { skipWhen: (state) => !state.confirmSetRetentionPeriodFileEnabled },
  );
```

```
);

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const confirmSetLegalHoldFileRetention = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetLegalHoldFileRetention",
    (state) =>
      `Would you like to add a legal hold to file0.txt in
    ${state.retentionBucketName}?`,
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setLegalHoldFileRetentionAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setLegalHoldFileRetentionAction",
    async (state) => {
      await client.send(
        new PutObjectLegalHoldCommand({
          Bucket: state.retentionBucketName,
          Key: "file0.txt",
          LegalHold: {
            Status: ObjectLockLegalHoldStatus.ON,
          },
        }),
      );
      console.log(
        `Modified legal hold for file0.txt in ${state.retentionBucketName}.`,
      );
    },
    { skipWhen: (state) => !state.confirmSetLegalHoldFileRetention },
  );

/**
 * @param {Scenarios} scenarios
 */
```

```

const confirmSetRetentionPeriodFileRetention = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetRetentionPeriodFileRetention",
    (state) =>
      `Would you like to add a 1 day Governance retention period to file1.txt in
      ${state.retentionBucketName}?
      Reminder: Only a user with the s3:BypassGovernanceRetention permission will be
      able to delete this file or its bucket until the retention period has expired.`
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setRetentionPeriodFileRetentionAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setRetentionPeriodFileRetentionAction",
    async (state) => {
      const retentionDate = new Date();
      retentionDate.setDate(retentionDate.getDate() + 1);
      await client.send(
        new PutObjectRetentionCommand({
          Bucket: state.retentionBucketName,
          Key: "file1.txt",
          Retention: {
            Mode: ObjectLockRetentionMode.GOVERNANCE,
            RetainUntilDate: retentionDate,
          },
          BypassGovernanceRetention: true,
        }),
      );
      console.log(
        `Set retention for file1.txt in ${state.retentionBucketName} until
        ${retentionDate.toISOString().split("T")[0]}.`,
      );
    },
    { skipWhen: (state) => !state.confirmSetRetentionPeriodFileRetention },
  );

export {
  createBuckets,

```

```
confirmCreateBuckets,  
createBucketsAction,  
populateBuckets,  
confirmPopulateBuckets,  
populateBucketsAction,  
updateRetention,  
confirmUpdateRetention,  
updateRetentionAction,  
updateLockPolicy,  
confirmUpdateLockPolicy,  
updateLockPolicyAction,  
confirmSetLegalHoldFileEnabled,  
setLegalHoldFileEnabledAction,  
confirmSetRetentionPeriodFileEnabled,  
setRetentionPeriodFileEnabledAction,  
confirmSetLegalHoldFileRetention,  
setLegalHoldFileRetentionAction,  
confirmSetRetentionPeriodFileRetention,  
setRetentionPeriodFileRetentionAction,  
};
```

repl.steps.js - Dateien in den Buckets anzeigen und löschen.

```
import {  
  ChecksumAlgorithm,  
  DeleteObjectCommand,  
  GetObjectLegalHoldCommand,  
  GetObjectLockConfigurationCommand,  
  GetObjectRetentionCommand,  
  ListObjectVersionsCommand,  
  PutObjectCommand,  
} from "@aws-sdk/client-s3";  
  
/**  
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios  
 */  
  
/**  
 * @typedef {import("@aws-sdk/client-s3").S3Client} S3Client  
 */  
  
const choices = {
```

```
EXIT: 0,
LIST_ALL_FILES: 1,
DELETE_FILE: 2,
DELETE_FILE_WITH_RETENTION: 3,
OVERWRITE_FILE: 4,
VIEW_RETENTION_SETTINGS: 5,
VIEW_LEGAL_HOLD_SETTINGS: 6,
};

/**
 * @param {Scenarios} scenarios
 */
const replInput = (scenarios) =>
  new scenarios.ScenarioInput(
    "replChoice",
    `Explore the S3 locking features by selecting one of the following choices`,
    {
      type: "select",
      choices: [
        { name: "List all files in buckets", value: choices.LIST_ALL_FILES },
        { name: "Attempt to delete a file.", value: choices.DELETE_FILE },
        {
          name: "Attempt to delete a file with retention period bypass.",
          value: choices.DELETE_FILE_WITH_RETENTION,
        },
        { name: "Attempt to overwrite a file.", value: choices.OVERWRITE_FILE },
        {
          name: "View the object and bucket retention settings for a file.",
          value: choices.VIEW_RETENTION_SETTINGS,
        },
        {
          name: "View the legal hold settings for a file.",
          value: choices.VIEW_LEGAL_HOLD_SETTINGS,
        },
        { name: "Finish the workflow.", value: choices.EXIT },
      ],
    },
  );

/**
 * @param {S3Client} client
 * @param {string[]} buckets
 */
const getAllFiles = async (client, buckets) => {
```

```
/** @type {{bucket: string, key: string, version: string}[]} */
const files = [];
for (const bucket of buckets) {
  const objectsResponse = await client.send(
    new ListObjectVersionsCommand({ Bucket: bucket }),
  );
  for (const version of objectsResponse.Versions || []) {
    const { Key, VersionId } = version;
    files.push({ bucket, key: Key, version: VersionId });
  }
}

return files;
};

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const replAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "replAction",
    async (state) => {
      const files = await getAllFiles(client, [
        state.noLockBucketName,
        state.lockEnabledBucketName,
        state.retentionBucketName,
      ]);

      const fileInput = new scenarios.ScenarioInput(
        "selectedFile",
        "Select a file:",
        {
          type: "select",
          choices: files.map((file, index) => ({
            name: `${index + 1}: ${file.bucket}: ${file.key} (version: ${
              file.version
            })`,
            value: index,
          })),
        },
      );

      const { replChoice } = state;
```

```
switch (replChoice) {
  case choices.LIST_ALL_FILES: {
    const files = await getAllFiles(client, [
      state.noLockBucketName,
      state.lockEnabledBucketName,
      state.retentionBucketName,
    ]);
    state.replOutput = files
      .map(
        (file) =>
          `${file.bucket}: ${file.key} (version: ${file.version})`,
      )
      .join("\n");
    break;
  }
  case choices.DELETE_FILE: {
    /** @type {number} */
    const fileToDelete = await fileInput.handle(state);
    const selectedFile = files[fileToDelete];
    try {
      await client.send(
        new DeleteObjectCommand({
          Bucket: selectedFile.bucket,
          Key: selectedFile.key,
          VersionId: selectedFile.version,
        }),
      );
      state.replOutput = `Deleted ${selectedFile.key} in
${selectedFile.bucket}.`;
    } catch (err) {
      state.replOutput = `Unable to delete object ${selectedFile.key} in
bucket ${selectedFile.bucket}: ${err.message}`;
    }
    break;
  }
  case choices.DELETE_FILE_WITH_RETENTION: {
    /** @type {number} */
    const fileToDelete = await fileInput.handle(state);
    const selectedFile = files[fileToDelete];
    try {
      await client.send(
        new DeleteObjectCommand({
          Bucket: selectedFile.bucket,
```

```
        Key: selectedFile.key,
        VersionId: selectedFile.version,
        BypassGovernanceRetention: true,
    )),
    );
    state.replOutput = `Deleted ${selectedFile.key} in
${selectedFile.bucket}.`;
    } catch (err) {
        state.replOutput = `Unable to delete object ${selectedFile.key} in
bucket ${selectedFile.bucket}: ${err.message}`;
    }
    break;
}
case choices.OVERWRITE_FILE: {
    /** @type {number} */
    const fileToOverwrite = await fileInput.handle(state);
    const selectedFile = files[fileToOverwrite];
    try {
        await client.send(
            new PutObjectCommand({
                Bucket: selectedFile.bucket,
                Key: selectedFile.key,
                Body: "New content",
                ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
            )),
        );
        state.replOutput = `Overwrote ${selectedFile.key} in
${selectedFile.bucket}.`;
    } catch (err) {
        state.replOutput = `Unable to overwrite object ${selectedFile.key} in
bucket ${selectedFile.bucket}: ${err.message}`;
    }
    break;
}
case choices.VIEW_RETENTION_SETTINGS: {
    /** @type {number} */
    const fileToView = await fileInput.handle(state);
    const selectedFile = files[fileToView];
    try {
        const retention = await client.send(
            new GetObjectRetentionCommand({
                Bucket: selectedFile.bucket,
                Key: selectedFile.key,
                VersionId: selectedFile.version,
```



```

    }),
  );
  const bucketConfig = await client.send(
    new GetObjectLockConfigurationCommand({
      Bucket: selectedFile.bucket,
    }),
  );
  state.replOutput = `Object retention for ${selectedFile.key}
in ${selectedFile.bucket}: ${retention.Retention?.Mode} until
${retention.Retention?.RetainUntilDate?.toISOString()}.
Bucket object lock config for ${selectedFile.bucket} in ${selectedFile.bucket}:
Enabled: ${bucketConfig.ObjectLockConfiguration?.ObjectLockEnabled}
Rule:
${JSON.stringify(bucketConfig.ObjectLockConfiguration?.Rule?.DefaultRetention)}`;
  } catch (err) {
    state.replOutput = `Unable to fetch object lock retention:
'${err.message}'`;
  }
  break;
}
case choices.VIEW_LEGAL_HOLD_SETTINGS: {
  /** @type {number} */
  const fileToView = await fileInput.handle(state);
  const selectedFile = files[fileToView];
  try {
    const legalHold = await client.send(
      new GetObjectLegalHoldCommand({
        Bucket: selectedFile.bucket,
        Key: selectedFile.key,
        VersionId: selectedFile.version,
      }),
    );
    state.replOutput = `Object legal hold for ${selectedFile.key} in
${selectedFile.bucket}: Status: ${legalHold.LegalHold?.Status}`;
  } catch (err) {
    state.replOutput = `Unable to fetch legal hold: '${err.message}'`;
  }
  break;
}
default:
  throw new Error(`Invalid replChoice: ${replChoice}`);
}
},
{

```

```

    whileConfig: {
      whileFn: ({ replChoice }) => replChoice !== choices.EXIT,
      input: replInput(scenarios),
      output: new scenarios.ScenarioOutput(
        "REPL output",
        (state) => state.replOutput,
        { preformatted: true },
      ),
    },
  },
},
);

export { replInput, replAction, choices };

```

clean.steps.js - Zerstöre alle erstellten Ressourcen.

```

import {
  DeleteObjectCommand,
  DeleteBucketCommand,
  ListObjectVersionsCommand,
  GetObjectLegalHoldCommand,
  GetObjectRetentionCommand,
  PutObjectLegalHoldCommand,
} from "@aws-sdk/client-s3";

/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
 */

/**
 * @typedef {import("@aws-sdk/client-s3").S3Client} S3Client
 */

/**
 * @param {Scenarios} scenarios
 */
const confirmCleanup = (scenarios) =>
  new scenarios.ScenarioInput("confirmCleanup", "Clean up resources?", {
    type: "confirm",
  });

/**

```

```
* @param {Scenarios} scenarios
* @param {S3Client} client
*/
const cleanupAction = (scenarios, client) =>
  new scenarios.ScenarioAction("cleanupAction", async (state) => {
    const { noLockBucketName, lockEnabledBucketName, retentionBucketName } =
      state;

    const buckets = [
      noLockBucketName,
      lockEnabledBucketName,
      retentionBucketName,
    ];

    for (const bucket of buckets) {
      /** @type {import("@aws-sdk/client-s3").ListObjectVersionsCommandOutput} */
      let objectsResponse;

      try {
        objectsResponse = await client.send(
          new ListObjectVersionsCommand({
            Bucket: bucket,
          }),
        );
      } catch (e) {
        if (e instanceof Error && e.name === "NoSuchBucket") {
          console.log("Object's bucket has already been deleted.");
          continue;
        } else {
          throw e;
        }
      }
    }

    for (const version of objectsResponse.Versions || []) {
      const { Key, VersionId } = version;

      try {
        const legalHold = await client.send(
          new GetObjectLegalHoldCommand({
            Bucket: bucket,
            Key,
            VersionId,
          }),
        );
      }
    }
  });
```

```
    if (legalHold.LegalHold?.Status === "ON") {
      await client.send(
        new PutObjectLegalHoldCommand({
          Bucket: bucket,
          Key,
          VersionId,
          LegalHold: {
            Status: "OFF",
          },
        }),
      );
    }
  } catch (err) {
    console.log(
      `Unable to fetch legal hold for ${Key} in ${bucket}:
    '${err.message}'`,
    );
  }

  try {
    const retention = await client.send(
      new GetObjectRetentionCommand({
        Bucket: bucket,
        Key,
        VersionId,
      }),
    );

    if (retention.Retention?.Mode === "GOVERNANCE") {
      await client.send(
        new DeleteObjectCommand({
          Bucket: bucket,
          Key,
          VersionId,
          BypassGovernanceRetention: true,
        }),
      );
    }
  } catch (err) {
    console.log(
      `Unable to fetch object lock retention for ${Key} in ${bucket}:
    '${err.message}'`,
    );
  }
}
```

```
    }

    await client.send(
      new DeleteObjectCommand({
        Bucket: bucket,
        Key,
        VersionId,
      }),
    );
  }

  await client.send(new DeleteBucketCommand({ Bucket: bucket }));
  console.log(`Delete for ${bucket} complete.`);
}
});

export { confirmCleanup, cleanupAction };
```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for JavaScript API-Referenz](#).
 - [GetObjectLegalHold](#)
 - [GetObjectLockConfiguration](#)
 - [GetObjectRetention](#)
 - [PutObjectLegalHold](#)
 - [PutObjectLockConfiguration](#)
 - [PutObjectRetention](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwalten Sie Zugriffskontrolllisten (ACLs) für Amazon S3 S3-Buckets mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie Zugriffskontrolllisten (ACLs) für Amazon S3 S3-Buckets verwaltet werden.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to manage Amazon Simple Storage Service
/// (Amazon S3) access control lists (ACLs) to control Amazon S3 bucket
/// access.
/// </summary>
public class ManageACLs
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket1";
        string newBucketName = "amzn-s3-demo-bucket2";
        string keyName = "sample-object.txt";
        string emailAddress = "someone@example.com";

        // If the AWS Region where your bucket is located is different from
        // the Region defined for the default user, pass the Amazon S3
bucket's
        // name to the client constructor. It should look like this:
        // RegionEndpoint bucketRegion = RegionEndpoint.USEast1;
        IAmazonS3 client = new AmazonS3Client();

        await TestBucketObjectACLsAsync(client, bucketName, newBucketName,
keyName, emailAddress);
    }

    /// <summary>
```

```
ACL    /// Creates a new Amazon S3 bucket with a canned ACL, then retrieves the
      /// information and then adds a new ACL to one of the objects in the
      /// Amazon S3 bucket.
      /// </summary>
      /// <param name="client">The initialized Amazon S3 client object used to
call    call
      /// methods to create a bucket, get an ACL, and add a different ACL to
      /// one of the objects.</param>
      /// <param name="bucketName">A string representing the original Amazon S3
      /// bucket name.</param>
      /// <param name="newBucketName">A string representing the name of the
      /// new bucket that will be created.</param>
      /// <param name="keyName">A string representing the key name of an Amazon
S3     S3
      /// object for which we will change the ACL.</param>
      /// <param name="emailAddress">A string representing the email address
      /// belonging to the person to whom access to the Amazon S3 bucket will
be     be
      /// granted.</param>
      public static async Task TestBucketObjectACLsAsync(
          IAmazonS3 client,
          string bucketName,
          string newBucketName,
          string keyName,
          string emailAddress)
      {
          try
          {
              // Create a new Amazon S3 bucket and specify canned ACL.
              var success = await CreateBucketWithCannedACLAsync(client,
newBucketName);
              // Get the ACL on a bucket.
              await GetBucketACLAsync(client, bucketName);
              // Add (replace) the ACL on an object in a bucket.
              await AddACLToExistingObjectAsync(client, bucketName, keyName,
emailAddress);
          }
          catch (AmazonS3Exception amazonS3Exception)
          {
              Console.WriteLine($"Exception: {amazonS3Exception.Message}");
          }
      }
```

```
    }

    /// <summary>
    /// Creates a new Amazon S3 bucket with a canned ACL attached.
    /// </summary>
    /// <param name="client">The initialized client object used to call
    /// PutBucketAsync.</param>
    /// <param name="newBucketName">A string representing the name of the
    /// new Amazon S3 bucket.</param>
    /// <returns>Returns a boolean value indicating success or failure.</
returns>
    public static async Task<bool> CreateBucketWithCannedACLAsync(IAmazonS3
client, string newBucketName)
    {
        var request = new PutBucketRequest()
        {
            BucketName = newBucketName,
            BucketRegion = S3Region.EUWest1,

            // Add a canned ACL.
            CannedACL = S3CannedACL.LogDeliveryWrite,
        };

        var response = await client.PutBucketAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }

    /// <summary>
    /// Retrieves the ACL associated with the Amazon S3 bucket name in the
    /// bucketName parameter.
    /// </summary>
    /// <param name="client">The initialized client object used to call
    /// PutBucketAsync.</param>
    /// <param name="bucketName">The Amazon S3 bucket for which we want to
get the
    /// ACL list.</param>
    /// <returns>Returns an S3AccessControlList returned from the call to
    /// GetACLAsync.</returns>
    public static async Task<S3AccessControlList> GetBucketACLAsync(IAmazonS3
client, string bucketName)
    {
        GetACLResponse response = await client.GetACLAsync(new GetACLRequest
        {
```



```

        BucketName = bucketName,
    });

    return response.AccessControlList;
}

/// <summary>
/// Adds a new ACL to an existing object in the Amazon S3 bucket.
/// </summary>
/// <param name="client">The initialized client object used to call
/// PutBucketAsync.</param>
/// <param name="bucketName">A string representing the name of the Amazon
S3
param> bucket containing the object to which we want to apply a new ACL.</
param>
/// <param name="keyName">A string representing the name of the object
/// to which we want to apply the new ACL.</param>
/// <param name="emailAddress">The email address of the person to whom
/// we will be applying to whom access will be granted.</param>
public static async Task AddACLToExistingObjectAsync(IAmazonS3 client,
string bucketName, string keyName, string emailAddress)
{
    // Retrieve the ACL for an object.
    GetACLResponse aclResponse = await client.GetACLAsync(new
GetACLRequest
    {
        BucketName = bucketName,
        Key = keyName,
    });

    S3AccessControlList acl = aclResponse.AccessControlList;

    // Retrieve the owner.
    Owner owner = acl.Owner;

    // Clear existing grants.
    acl.Grants.Clear();

    // Add a grant to reset the owner's full permission
    // (the previous clear statement removed all permissions).
    var fullControlGrant = new S3Grant
    {

```

```
        Grantee = new S3Grantee { CanonicalUser = acl.Owner.Id },
    };
    acl.AddGrant(fullControlGrant.Grantee, S3Permission.FULL_CONTROL);

    // Specify email to identify grantee for granting permissions.
    var grantUsingEmail = new S3Grant
    {
        Grantee = new S3Grantee { EmailAddress = emailAddress },
        Permission = S3Permission.WRITE_ACP,
    };

    // Specify log delivery group as grantee.
    var grantLogDeliveryGroup = new S3Grant
    {
        Grantee = new S3Grantee { URI = "http://acs.amazonaws.com/groups/
s3/LogDelivery" },
        Permission = S3Permission.WRITE,
    };

    // Create a new ACL.
    var newAcl = new S3AccessControlList
    {
        Grants = new List<S3Grant> { grantUsingEmail,
grantLogDeliveryGroup },
        Owner = owner,
    };

    // Set the new ACL. We're throwing away the response here.
    _ = await client.PutACLAsync(new PutACLRequest
    {
        BucketName = bucketName,
        Key = keyName,
        AccessControlList = newAcl,
    });
}
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK for .NET API-Referenz.
- [GetBucketAcl](#)

- [GetObjectAcl](#)
- [PutBucketAcl](#)
- [PutObjectAcl](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwalten Sie versionierte Amazon S3 S3-Objekte stapelweise mit einer Lambda-Funktion mithilfe eines AWS SDK

Das folgende Codebeispiel zeigt, wie versionierte S3-Objekte in Batches mit einer Lambda-Funktion verwaltet werden.

Python

SDK für Python (Boto3)

Zeigt, wie versionierte Objekte von Amazon Simple Storage Service (Amazon S3) stapelweise bearbeitet werden, indem Aufträge erstellt werden, die AWS Lambda-Funktionen zur Durchführung der Verarbeitung. In diesem Beispiel wird ein versionsfähiger Bucket erstellt, der die Strophen des Gedichts *You Are Old, Father William* von Lewis Carroll hochlädt und Amazon-S3-Batch-Aufträge verwendet, um das Gedicht auf verschiedene Arten zu verdrehen.

Lernen Sie Folgendes:

- Erstellen Sie Lambda-Funktionen, die mit versionierten Objekten arbeiten.
- Erstellen Sie ein Manifest von zu aktualisierenden Objekten.
- Erstellen Sie Batch-Aufträge, die Lambda-Funktionen zum Aktualisieren von Objekten aufrufen.
- Löschen Sie Lambda-Funktionen.
- Leeren und löschen Sie einen versionierten Bucket.

Dieses Beispiel lässt sich am besten auf [ansehen GitHub](#). Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon S3

Eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Analysieren Sie Amazon S3 URIs mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie Amazon S3 analysiert wird, URIs um wichtige Komponenten wie den Bucket-Namen und den Objektschlüssel zu extrahieren.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Analysieren Sie einen Amazon S3 URI mithilfe der [S3Uri-Klasse](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.S3Uri;
import software.amazon.awssdk.services.s3.S3Utilities;

import java.net.URI;
import java.util.List;
import java.util.Map;

/**
 *
 * @param s3Client - An S3Client through which you acquire an S3Uri
instance.
 * @param s3ObjectUrl - A complex URL (String) that is used to demonstrate
S3Uri
```

```

*           capabilities.
*/
public static void parseS3UriExample(S3Client s3Client, String s3objectUrl) {
    logger.info(s3objectUrl);
    // Console output:
    // 'https://s3.us-west-1.amazonaws.com/myBucket/resources/doc.txt?
versionId=abc123&partNumber=77&partNumber=88'.

    // Create an S3Utilities object using the configuration of the s3Client.
    S3Utilities s3Utilities = s3Client.utilities();

    // From a String URL create a URI object to pass to the parseUri()
method.
    URI uri = URI.create(s3objectUrl);
    S3Uri s3Uri = s3Utilities.parseUri(uri);

    // If the URI contains no value for the Region, bucket or key, the SDK
returns
    // an empty Optional.
    // The SDK returns decoded URI values.

    Region region = s3Uri.region().orElse(null);
    log("region", region);
    // Console output: 'region: us-west-1'.

    String bucket = s3Uri.bucket().orElse(null);
    log("bucket", bucket);
    // Console output: 'bucket: myBucket'.

    String key = s3Uri.key().orElse(null);
    log("key", key);
    // Console output: 'key: resources/doc.txt'.

    Boolean isPathStyle = s3Uri.isPathStyle();
    log("isPathStyle", isPathStyle);
    // Console output: 'isPathStyle: true'.

    // If the URI contains no query parameters, the SDK returns an empty map.
    Map<String, List<String>> queryParams = s3Uri.rawQueryParameters();
    log("rawQueryParameters", queryParams);
    // Console output: 'rawQueryParameters: {versionId=[abc123],
partNumber=[77,
    // 88]}'.

```

```
// Retrieve the first or all values for a query parameter as shown in the
// following code.
String versionId =
s3Uri.firstMatchingRawQueryParameter("versionId").orElse(null);
log("firstMatchingRawQueryParameter-versionId", versionId);
// Console output: 'firstMatchingRawQueryParameter-versionId: abc123'.

String partNumber =
s3Uri.firstMatchingRawQueryParameter("partNumber").orElse(null);
log("firstMatchingRawQueryParameter-partNumber", partNumber);
// Console output: 'firstMatchingRawQueryParameter-partNumber: 77'.

List<String> partNumbers =
s3Uri.firstMatchingRawQueryParameters("partNumber");
log("firstMatchingRawQueryParameter", partNumbers);
// Console output: 'firstMatchingRawQueryParameter: [77, 88]'.

/*
 * Object keys and query parameters with reserved or unsafe characters,
must be
 * URL-encoded.
 * For example replace whitespace " " with "%20".
 * Valid:
 * "https://s3.us-west-1.amazonaws.com/myBucket/object%20key?query=
%5Bbrackets%5D"
 * Invalid:
 * "https://s3.us-west-1.amazonaws.com/myBucket/object key?
query=[brackets]"
 *
 * Virtual-hosted-style URIs with bucket names that contain a dot, ".",
the dot
 * must not be URL-encoded.
 * Valid: "https://my.Bucket.s3.us-west-1.amazonaws.com/key"
 * Invalid: "https://my%2EBucket.s3.us-west-1.amazonaws.com/key"
 */
}

private static void log(String s3UriElement, Object element) {
    if (element == null) {
        logger.info("{}: {}", s3UriElement, "null");
    } else {
        logger.info("{}: {}", s3UriElement, element);
    }
}
}
```

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erstellen Sie eine mehrteilige Kopie eines Amazon S3 S3-Objekts mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie Sie eine mehrteilige Kopie eines Amazon-S3-Objekts erstellen.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to perform a multi-part copy from one Amazon
/// Simple Storage Service (Amazon S3) bucket to another.
/// </summary>
public class MPUapiCopyObj
{
    private const string SourceBucket = "amzn-s3-demo-bucket1";
    private const string TargetBucket = "amzn-s3-demo-bucket2";
    private const string SourceObjectKey = "example.mov";
    private const string TargetObjectKey = "copied_video_file.mov";

    /// <summary>
    /// This method starts the multi-part upload.
```

```
/// </summary>
public static async Task Main()
{
    var s3Client = new AmazonS3Client();
    Console.WriteLine("Copying object...");
    await MPUCopyObjectAsync(s3Client);
}

/// <summary>
/// This method uses the passed client object to perform a multipart
/// copy operation.
/// </summary>
/// <param name="client">An Amazon S3 client object that will be used
/// to perform the copy.</param>
public static async Task MPUCopyObjectAsync(AmazonS3Client client)
{
    // Create a list to store the copy part responses.
    var copyResponses = new List<CopyPartResponse>();

    // Setup information required to initiate the multipart upload.
    var initiateRequest = new InitiateMultipartUploadRequest
    {
        BucketName = TargetBucket,
        Key = TargetObjectKey,
    };

    // Initiate the upload.
    InitiateMultipartUploadResponse initResponse =
        await client.InitiateMultipartUploadAsync(initiateRequest);

    // Save the upload ID.
    string uploadId = initResponse.UploadId;

    try
    {
        // Get the size of the object.
        var metadataRequest = new GetObjectMetadataRequest
        {
            BucketName = SourceBucket,
            Key = SourceObjectKey,
        };

        GetObjectMetadataResponse metadataResponse =
            await client.GetObjectMetadataAsync(metadataRequest);
    }
}
```



```
var objectSize = metadataResponse.ContentLength; // Length in
bytes.

// Copy the parts.
var partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

long bytePosition = 0;
for (int i = 1; bytePosition < objectSize; i++)
{
    var copyRequest = new CopyPartRequest
    {
        DestinationBucket = TargetBucket,
        DestinationKey = TargetObjectKey,
        SourceBucket = SourceBucket,
        SourceKey = SourceObjectKey,
        UploadId = uploadId,
        FirstByte = bytePosition,
        LastByte = bytePosition + partSize - 1 >= objectSize ?
objectSize - 1 : bytePosition + partSize - 1,
        PartNumber = i,
    };

    copyResponses.Add(await client.CopyPartAsync(copyRequest));

    bytePosition += partSize;
}

// Set up to complete the copy.
var completeRequest = new CompleteMultipartUploadRequest
{
    BucketName = TargetBucket,
    Key = TargetObjectKey,
    UploadId = initResponse.UploadId,
};
completeRequest.AddPartETags(copyResponses);

// Complete the copy.
CompleteMultipartUploadResponse completeUploadResponse =
    await client.CompleteMultipartUploadAsync(completeRequest);
}
catch (AmazonS3Exception e)
{
    Console.WriteLine($"Error encountered on server.
Message: '{e.Message}' when writing an object");
}
```

```
    }
    catch (Exception e)
    {
        Console.WriteLine($"Unknown encountered on server.
Message: '{e.Message}' when writing an object");
    }
}
}
```

- API-Einheiten finden Sie in den folgenden Themen unter AWS SDK for .NET API-Referenz.
 - [CompleteMultipartUpload](#)
 - [CreateMultipartUpload](#)
 - [GetObjectMetadata](#)
 - [UploadPartCopy](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Führen Sie einen mehrteiligen Upload eines Amazon S3 S3-Objekts mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie Sie einen mehrteiligen Upload in ein Amazon-S3-Objekt durchführen.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

In den Codebeispielen werden folgende Importe verwendet.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;

import java.io.IOException;
import java.io.RandomAccessFile;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.ByteBuffer;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;
```

Verwenden Sie den [S3 Transfer Manager](#) zusätzlich zu [AWS CRT-basierter S3-Client](#) zur transparenten Durchführung eines mehrteiligen Uploads, wenn die Größe des Inhalts einen Schwellenwert überschreitet. Der Standardschwellenwert beträgt 8 MB.

```
public void multipartUploadWithTransferManager(String filePath) {
    S3TransferManager transferManager = S3TransferManager.create();
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b
            .bucket(bucketName)
            .key(key))
        .source(Paths.get(filePath))
        .build();
```

```
FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
fileUpload.completionFuture().join();
transferManager.close();
}
```

Verwenden Sie den [S3Client, um einen mehrteiligen API Upload](#) durchzuführen.

```
public void multipartUploadWithS3Client(String filePath) {

    // Initiate the multipart upload.
    CreateMultipartUploadResponse createMultipartUploadResponse =
s3Client.createMultipartUpload(b -> b
        .bucket(bucketName)
        .key(key));
    String uploadId = createMultipartUploadResponse.uploadId();

    // Upload the parts of the file.
    int partNumber = 1;
    List<CompletedPart> completedParts = new ArrayList<>();
    ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

    try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
        long fileSize = file.length();
        long position = 0;
        while (position < fileSize) {
            file.seek(position);
            long read = file.getChannel().read(bb);

            bb.flip(); // Swap position and limit before reading from the
buffer.

            UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                .bucket(bucketName)
                .key(key)
                .uploadId(uploadId)
                .partNumber(partNumber)
                .build();

            UploadPartResponse partResponse = s3Client.uploadPart(
                uploadPartRequest,
                RequestBody.fromByteBuffer(bb));

            CompletedPart part = CompletedPart.builder()
```

```

        .partNumber(partNumber)
        .eTag(partResponse.eTag())
        .build();
    completedParts.add(part);

    bb.clear();
    position += read;
    partNumber++;
}
} catch (IOException e) {
    logger.error(e.getMessage());
}

// Complete the multipart upload.
s3Client.completeMultipartUpload(b -> b
    .bucket(bucketName)
    .key(key)
    .uploadId(uploadId)

.multipartUpload(CompletedMultipartUpload.builder().parts(completedParts).build()));
}

```

Verwenden Sie den [S3 AsyncClient API](#) mit aktivierter Multipart-Support, um einen mehrteiligen Upload durchzuführen.

```

public void multipartUploadWithS3AsyncClient(String filePath) {
    // Enable multipart support.
    S3AsyncClient s3AsyncClient = S3AsyncClient.builder()
        .multipartEnabled(true)
        .build();

    CompletableFuture<PutObjectResponse> response = s3AsyncClient.putObject(b
-> b
        .bucket(bucketName)
        .key(key),
        Paths.get(filePath));

    response.join();
    logger.info("File uploaded in multiple 8 MiB parts using
S3AsyncClient.");
}

```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for Java 2.x API-Referenz](#).
 - [CompleteMultipartUpload](#)
 - [CreateMultipartUpload](#)
 - [UploadPart](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Empfangen und verarbeiten Sie Amazon S3 S3-Ereignisbenachrichtigungen mithilfe eines AWS SDK.

Das folgende Codebeispiel zeigt, wie Sie objektorientiert mit S3-Ereignisbenachrichtigungen arbeiten.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Dieses Beispiel zeigt, wie ein S3-Benachrichtigungsereignis mithilfe von Amazon verarbeitet wird.

```
/**
 * This method receives S3 event notifications by using an SqsAsyncClient.
 * After the client receives the messages it deserializes the JSON payload
and logs them. It uses
 * the S3EventNotification class (part of the S3 event notification API for
Java) to deserialize
 * the JSON payload and access the messages in an object-oriented way.
 *

```

```
* @param queueUrl The URL of the AWS SQS queue that receives the S3 event
notifications.
* @see <a href="https://sdk.amazonaws.com/java/api/latest/software.amazon/
awssdk/eventnotifications/s3/model/package-summary.html">S3EventNotification
API</a>.
* <p>
* To use S3 event notification serialization/deserialization to objects, add
the following
* dependency to your Maven pom.xml file.
* <dependency>
* <groupId>software.amazon.awssdk</groupId>
* <artifactId>s3-event-notifications</artifactId>
* <version><LATEST></version>
* </dependency>
* <p>
* The S3 event notification API became available with version 2.25.11 of the
Java SDK.
* <p>
* This example shows the use of the API with AWS SQS, but it can be used to
process S3 event notifications
* in AWS SNS or AWS Lambda as well.
* <p>
* Note: The S3EventNotification class does not work with messages routed
through AWS EventBridge.
*/
static void processS3Events(String bucketName, String queueUrl, String
queueArn) {
    try {
        // Configure the bucket to send Object Created and Object Tagging
notifications to an existing SQS queue.
        s3Client.putBucketNotificationConfiguration(b -> b
            .notificationConfiguration(ncb -> ncb
                .queueConfigurations(qcb -> qcb
                    .events(Event.S3_OBJECT_CREATED,
Event.S3_OBJECT_TAGGING)
                        .queueArn(queueArn)))
                .bucket(bucketName)
            ).join();

        triggerS3EventNotifications(bucketName);
        // Wait for event notifications to propagate.
        Thread.sleep(Duration.ofSeconds(5).toMillis());

        boolean didReceiveMessages = true;
```

```
        while (didReceiveMessages) {
            // Display the number of messages that are available in the
queue.
            sqsClient.getQueueAttributes(b -> b
                .queueUrl(queueUrl)
                .attributeNames(QueueAttributeName.APPROXIMATE_NUMBER_OF_MESSAGES)
                ).thenAccept(attributeResponse ->
                logger.info("Approximate number of messages in
the queue: {}"),
attributeResponse.attributes().get(QueueAttributeName.APPROXIMATE_NUMBER_OF_MESSAGES)))
                .join();

            // Receive the messages.
            ReceiveMessageResponse response = sqsClient.receiveMessage(b -> b
                .queueUrl(queueUrl)
                ).get();
            logger.info("Count of received messages: {}",
response.messages().size());
            didReceiveMessages = !response.messages().isEmpty();

            // Create a collection to hold the received message for deletion
            // after we log the messages.
            HashSet<DeleteMessageBatchRequestEntry> messagesToDelete = new
HashSet<>();
            // Process each message.
            response.messages().forEach(message -> {
                logger.info("Message id: {}", message.messageId());
                // Deserialize JSON message body to a S3EventNotification
object
                // to access messages in an object-oriented way.
                S3EventNotification event =
S3EventNotification.fromJson(message.body());

                // Log the S3 event notification record details.
                if (event.getRecords() != null) {
                    event.getRecords().forEach(record -> {
                        String eventName = record.getEventName();
                        String key = record.getS3().getObject().getKey();
                        logger.info(record.toString());
                        logger.info("Event name is {} and key is {}",
eventName, key);
                    });
                }
            });
        }
    }
}
```



```
        }
        // Add logged messages to collection for batch deletion.
        messagesToDelete.add(DeleteMessageBatchRequestEntry.builder()
            .id(message.getMessageId())
            .receiptHandle(message.getReceiptHandle())
            .build());
    });
    // Delete messages.
    if (!messagesToDelete.isEmpty()) {

        sqsClient.deleteMessageBatch(DeleteMessageBatchRequest.builder()
            .queueUrl(queueUrl)
            .entries(messagesToDelete)
            .build()
        ).join();
    }
    } // End of while block.
} catch (InterruptedException | ExecutionException e) {
    throw new RuntimeException(e);
}
}
```

- APIEinzelheiten finden Sie in den folgenden Themen unter AWS SDK for Java 2.x APIReferenz.
 - [DeleteMessageBatch](#)
 - [GetQueueAttributes](#)
 - [PutBucketNotificationConfiguration](#)
 - [ReceiveMessage](#)

Für eine vollständige Liste von AWS SDKEntwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Speichern EXIF und andere Bildinformationen mithilfe eines AWS SDK

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Ruft EXIF Informationen aus einer A- JPGJPEG, oder PNG -Datei ab.

- Laden Sie die Bilddatei in einen Amazon-S3-Bucket hoch.
- Verwenden Sie Amazon Rekognition, um die drei wichtigsten Attribute (Labels) in der Datei zu identifizieren.
- Fügen Sie die Informationen EXIF und das Etikett zu einer Amazon DynamoDB-Tabelle in der Region hinzu.

Rust

SDK für Rust

Rufen Sie EXIF Informationen aus einer JPG, JPEG, oder PNG -Datei ab, laden Sie die Bilddatei in einen Amazon S3 S3-Bucket hoch, verwenden Sie Amazon Rekognition, um die drei wichtigsten Attribute (Labels in Amazon Rekognition) in der Datei zu identifizieren, und fügen Sie die EXIF und Labelinformationen zu einer Amazon DynamoDB-Tabelle in der Region hinzu.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter. [GitHub](#)

In diesem Beispiel verwendete Dienste

- DynamoDB
- Amazon Rekognition
- Amazon S3

Eine vollständige Liste von AWS SDK Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Senden Sie S3-Ereignisbenachrichtigungen an Amazon EventBridge mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie ein Bucket aktiviert wird, um S3-Ereignisbenachrichtigungen an ein SNS Amazon-Thema EventBridge und eine SQS Amazon-Warteschlange zu senden und Benachrichtigungen an diese weiterzuleiten.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
/** This method configures a bucket to send events to AWS EventBridge and
creates a rule
 * to route the S3 object created events to a topic and a queue.
 *
 * @param bucketName Name of existing bucket
 * @param topicArn ARN of existing topic to receive S3 event notifications
 * @param queueArn ARN of existing queue to receive S3 event notifications
 *
 * An AWS CloudFormation stack sets up the bucket, queue, topic before the
method runs.
 */
public static String setBucketNotificationToEventBridge(String bucketName,
String topicArn, String queueArn) {
    try {
        // Enable bucket to emit S3 Event notifications to EventBridge.
        s3Client.putBucketNotificationConfiguration(b -> b
            .bucket(bucketName)
            .notificationConfiguration(b1 -> b1
                .eventBridgeConfiguration(
                    SdkBuilder::build)
            ).build()).join();

        // Create an EventBridge rule to route Object Created notifications.
        PutRuleRequest putRuleRequest = PutRuleRequest.builder()
            .name(RULE_NAME)
            .eventPattern("""
                {
                    "source": ["aws.s3"],
                    "detail-type": ["Object Created"],
                    "detail": {
                        "bucket": {
                            "name": ["%s"]
                        }
                    }
                }
            """)
            .build();
    } catch (Exception e) {
        // Handle exception
    }
}
```

```

        }
    }
}
"".formatted(bucketName))
.build();

// Add the rule to the default event bus.
PutRuleResponse putRuleResponse =
eventBridgeClient.putRule(putRuleRequest)
    .whenComplete((r, t) -> {
        if (t != null) {
            logger.error("Error creating event bus rule: " +
t.getMessage(), t);
                throw new RuntimeException(t.getCause().getMessage(),
t);
        }
        logger.info("Event bus rule creation request sent
successfully. ARN is: {}", r.ruleArn());
    }).join();

// Add the existing SNS topic and SQS queue as targets to the rule.
eventBridgeClient.putTargets(b -> b
    .eventBusName("default")
    .rule(RULE_NAME)
    .targets(List.of (
        Target.builder()
            .arn(queueArn)
            .id("Queue")
            .build(),
        Target.builder()
            .arn(topicArn)
            .id("Topic")
            .build()
    )
    ).join();
    return putRuleResponse.ruleArn();
} catch (S3Exception e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return null;
}

```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for Java 2.x API-Referenz](#).
- [PutBucketNotificationConfiguration](#)
- [PutRule](#)
- [PutTargets](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verfolgen Sie den Upload oder Download eines Amazon S3 S3-Objekts mithilfe eines AWS SDK

Das folgende Codebeispiel zeigt, wie Sie den Upload oder Download eines Amazon S3 S3-Objekts verfolgen können.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Verfolgen Sie den Fortschritt eines Datei-Uploads.

```
public void trackUploadFile(S3TransferManager transferManager, String
bucketName,
                        String key, URI filePathURI) {
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b.bucket(bucketName).key(key))
        .addTransferListener(LoggingTransferListener.create()) // Add
listener.
        .source(Paths.get(filePathURI))
        .build();

    FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
```

```

fileUpload.completionFuture().join();
/*
    The SDK provides a LoggingTransferListener implementation of the
TransferListener interface.
    You can also implement the interface to provide your own logic.

    Configure log4J2 with settings such as the following.
    <Configuration status="WARN">
        <Appenders>
            <Console name="AlignedConsoleAppender"
target="SYSTEM_OUT">
                <PatternLayout pattern="%m%n"/>
            </Console>
        </Appenders>

        <Loggers>
            <logger
name="software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener"
level="INFO" additivity="false">
                <AppenderRef ref="AlignedConsoleAppender"/>
            </logger>
        </Loggers>
    </Configuration>

    Log4J2 logs the progress. The following is example output for a 21.3
MB file upload.
    Transfer initiated...
    |                               | 0.0%
    |====                          | 21.1%
    |=====                        | 60.5%
    |=====                        | 100.0%
    Transfer complete!

*/
}

```

Verfolgen Sie den Fortschritt eines Datei-Downloads.

```

public void trackDownloadFile(S3TransferManager transferManager, String
bucketName,
                                String key, String downloadedFilePath) {
    DownloadFileRequest downloadFileRequest = DownloadFileRequest.builder()

```

```

        .getObjectRequest(b -> b.bucket(bucketName).key(key))
        .addTransferListener(LoggingTransferListener.create()) // Add
listener.

        .destination(Paths.get(downloadedFilePath))
        .build();

```

```

    FileDownload downloadFile =
transferManager.downloadFile(downloadFileRequest);

```

```

    CompletedFileDownload downloadResult =
downloadFile.completionFuture().join();
    /*

```

The SDK provides a `LoggingTransferListener` implementation of the `TransferListener` interface.

You can also implement the interface to provide your own logic.

Configure `log4j2` with settings such as the following.

```

<Configuration status="WARN">
  <Appenders>
    <Console name="AlignedConsoleAppender"
target="SYSTEM_OUT">
      <PatternLayout pattern="%m%n"/>
    </Console>
  </Appenders>

  <Loggers>
    <logger
name="software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener"
level="INFO" additivity="false">
      <AppenderRef ref="AlignedConsoleAppender"/>
    </logger>
  </Loggers>
</Configuration>

```

`Log4j2` logs the progress. The following is example output for a 21.3 MB file download.

```

Transfer initiated...
|=====          | 39.4%
|=====          | 78.8%
|=====          | 100.0%
Transfer complete!

```

```

    */
}

```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for Java 2.x API-Referenz](#).
 - [GetObject](#)
 - [PutObject](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Transformieren Sie Daten für Ihre Anwendung mit S3 Object Lambda

Das folgende Codebeispiel zeigt, wie Sie Daten für Ihre Anwendung mit S3 Object Lambda transformieren.

.NET

AWS SDK for .NET

Zeigt, wie benutzerdefinierten Code zu standardmäßigen GET S3-Anfragen hinzugefügt wird, um das von S3 abgerufene angeforderte Objekt so zu ändern, dass das Objekt den Anforderungen des anfordernden Clients oder der Anwendung entspricht.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Lambda
- Amazon S3

Eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Beispielansätze für Einheiten- und Integrationstests mit einem AWS SDK

Das folgende Codebeispiel zeigt Beispiele für bewährte Techniken beim Schreiben von Einheiten- und Integrationstests mit einem AWS SDK.

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Cargo.toml für Testbeispiele

```
[package]
name = "testing-examples"
version = "0.1.0"
authors = [
  "John Disanti <jdisanti@amazon.com>",
  "Doug Schwartz <dougsch@amazon.com>",
]
edition = "2021"

[dependencies]
async-trait = "0.1.51"
aws-config = { version = "1.0.1", features = ["behavior-version-latest"] }
aws-credential-types = { version = "1.0.1", features = [ "hardcoded-credentials", ] }
aws-sdk-s3 = { version = "1.4.0" }
aws-smithy-types = { version = "1.0.1" }
aws-smithy-runtime = { version = "1.0.1", features = ["test-util"] }
aws-smithy-runtime-api = { version = "1.0.1", features = ["test-util"] }
aws-types = { version = "1.0.1" }
clap = { version = "~4.4", features = ["derive"] }
http = "0.2.9"
mockall = "0.11.4"
serde_json = "1"
tokio = { version = "1.20.1", features = ["full"] }
tracing-subscriber = { version = "0.3.15", features = ["env-filter"] }
```

```
[[bin]]
name = "main"
path = "src/main.rs"
```

Beispiel für Modultests mit automock und einem Service-Wrapper

```
use aws_sdk_s3 as s3;
#[allow(unused_imports)]
use mockall::automock;

use s3::operation::list_objects_v2::{ListObjectsV2Error, ListObjectsV2Output};

#[cfg(test)]
pub use MockS3Impl as S3;
#[cfg(not(test))]
pub use S3Impl as S3;

#[allow(dead_code)]
pub struct S3Impl {
    inner: s3::Client,
}

#[cfg_attr(test, automock)]
impl S3Impl {
    #[allow(dead_code)]
    pub fn new(inner: s3::Client) -> Self {
        Self { inner }
    }

    #[allow(dead_code)]
    pub async fn list_objects(
        &self,
        bucket: &str,
        prefix: &str,
        continuation_token: Option<String>,
    ) -> Result<ListObjectsV2Output, s3::error::SdkError<ListObjectsV2Error>> {
        self.inner
            .list_objects_v2()
            .bucket(bucket)
            .prefix(prefix)
    }
}
```

```
        .set_continuation_token(continuation_token)
        .send()
        .await
    }
}

#[allow(dead_code)]
pub async fn determine_prefix_file_size(
    // Now we take a reference to our trait object instead of the S3 client
    // s3_list: ListObjectsService,
    s3_list: S3,
    bucket: &str,
    prefix: &str,
) -> Result<usize, s3::Error> {
    let mut next_token: Option<String> = None;
    let mut total_size_bytes = 0;
    loop {
        let result = s3_list
            .list_objects(bucket, prefix, next_token.take())
            .await?;

        // Add up the file sizes we got back
        for object in result.contents() {
            total_size_bytes += object.size().unwrap_or(0) as usize;
        }

        // Handle pagination, and break the loop if there are no more pages
        next_token = result.next_continuation_token.clone();
        if next_token.is_none() {
            break;
        }
    }
    Ok(total_size_bytes)
}

#[cfg(test)]
mod test {
    use super::*;
    use mockall::predicate::eq;

    #[tokio::test]
    async fn test_single_page() {
        let mut mock = MockS3Impl::default();
        mock.expect_list_objects()
    }
}
```

```
.with(eq("test-bucket"), eq("test-prefix"), eq(None))
.return_once(|_, _, _| {
    Ok(ListObjectsV2Output::builder()
        .set_contents(Some(vec![
            // Mock content for ListObjectsV2 response
            s3::types::Object::builder().size(5).build(),
            s3::types::Object::builder().size(2).build(),
        ]))
        .build())
});

// Run the code we want to test with it
let size = determine_prefix_file_size(mock, "test-bucket", "test-prefix")
    .await
    .unwrap();

// Verify we got the correct total size back
assert_eq!(7, size);
}

#[tokio::test]
async fn test_multiple_pages() {
    // Create the Mock instance with two pages of objects now
    let mut mock = MockS3Impl::default();
    mock.expect_list_objects()
        .with(eq("test-bucket"), eq("test-prefix"), eq(None))
        .return_once(|_, _, _| {
            Ok(ListObjectsV2Output::builder()
                .set_contents(Some(vec![
                    // Mock content for ListObjectsV2 response
                    s3::types::Object::builder().size(5).build(),
                    s3::types::Object::builder().size(2).build(),
                ]))
                .set_next_continuation_token(Some("next".to_string()))
                .build())
        });
    mock.expect_list_objects()
        .with(
            eq("test-bucket"),
            eq("test-prefix"),
            eq(Some("next".to_string())),
        )
        .return_once(|_, _, _| {
            Ok(ListObjectsV2Output::builder()
                .set_contents(Some(vec![
                    // Mock content for ListObjectsV2 response
                    s3::types::Object::builder().size(5).build(),
                    s3::types::Object::builder().size(2).build(),
                ]))
                .set_next_continuation_token(Some("next".to_string()))
                .build())
        });
}
```

```

        .set_contents(Some(vec![
            // Mock content for ListObjectsV2 response
            s3::types::Object::builder().size(3).build(),
            s3::types::Object::builder().size(9).build(),
        ]))
        .build()
    });

    // Run the code we want to test with it
    let size = determine_prefix_file_size(mock, "test-bucket", "test-prefix")
        .await
        .unwrap();

    assert_eq!(19, size);
}
}

```

Beispiel für Integrationstests mit StaticReplayClient.

```

use aws_sdk_s3 as s3;

#[allow(dead_code)]
pub async fn determine_prefix_file_size(
    // Now we take a reference to our trait object instead of the S3 client
    // s3_list: ListObjectsService,
    s3: s3::Client,
    bucket: &str,
    prefix: &str,
) -> Result<usize, s3::Error> {
    let mut next_token: Option<String> = None;
    let mut total_size_bytes = 0;
    loop {
        let result = s3
            .list_objects_v2()
            .prefix(prefix)
            .bucket(bucket)
            .set_continuation_token(next_token.take())
            .send()
            .await?;

        // Add up the file sizes we got back
    }
}

```

```
    for object in result.contents() {
        total_size_bytes += object.size().unwrap_or(0) as usize;
    }

    // Handle pagination, and break the loop if there are no more pages
    next_token = result.next_continuation_token.clone();
    if next_token.is_none() {
        break;
    }
}
Ok(total_size_bytes)
}

#[allow(dead_code)]
fn make_s3_test_credentials() -> s3::config::Credentials {
    s3::config::Credentials::new(
        "ATESTCLIENT",
        "astestsecretkey",
        Some("atestsessiontoken".to_string()),
        None,
        "",
    )
}

#[cfg(test)]
mod test {
    use super::*;
    use aws_config::BehaviorVersion;
    use aws_sdk_s3 as s3;
    use aws_smithy_runtime::client::http::test_util::{ReplayEvent,
StaticReplayClient};
    use aws_smithy_types::body::SdkBody;

    #[tokio::test]
    async fn test_single_page() {
        let page_1 = ReplayEvent::new(
            http::Request::builder()
                .method("GET")
                .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix")
                .body(SdkBody::empty())
                .unwrap(),
            http::Response::builder()
                .status(200)

```

```
        .body(SdkBody::from(include_str!("./testing/
response_1.xml")))
        .unwrap(),
    );
    let replay_client = StaticReplayClient::new(vec![page_1]);
    let client: s3::Client = s3::Client::from_conf(
        s3::Config::builder()
            .behavior_version(BehaviorVersion::latest())
            .credentials_provider(make_s3_test_credentials())
            .region(s3::config::Region::new("us-east-1"))
            .http_client(replay_client.clone())
            .build(),
    );

    // Run the code we want to test with it
    let size = determine_prefix_file_size(client, "test-bucket", "test-
prefix")
        .await
        .unwrap();

    // Verify we got the correct total size back
    assert_eq!(7, size);
    replay_client.assert_requests_match(&[]);
}

#[tokio::test]
async fn test_multiple_pages() {
    let page_1 = ReplayEvent::new(
        http::Request::builder()
            .method("GET")
            .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix")
            .body(SdkBody::empty())
            .unwrap(),
        http::Response::builder()
            .status(200)
            .body(SdkBody::from(include_str!("./testing/
response_multi_1.xml")))
            .unwrap(),
    );
    let page_2 = ReplayEvent::new(
        http::Request::builder()
            .method("GET")
```

```

        .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix&continuation-token=next")
        .body(SdkBody::empty())
        .unwrap(),
        http::Response::builder()
        .status(200)
        .body(SdkBody::from(include_str!("./testing/
response_multi_2.xml")))
        .unwrap(),
    );
    let replay_client = StaticReplayClient::new(vec![page_1, page_2]);
    let client: s3::Client = s3::Client::from_conf(
        s3::Config::builder()
        .behavior_version(BehaviorVersion::latest())
        .credentials_provider(make_s3_test_credentials())
        .region(s3::config::Region::new("us-east-1"))
        .http_client(replay_client.clone())
        .build(),
    );

    // Run the code we want to test with it
    let size = determine_prefix_file_size(client, "test-bucket", "test-
prefix")
        .await
        .unwrap();

    assert_eq!(19, size);

    replay_client.assert_requests_match(&[]);
}
}

```

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Rekursives Hochladen eines lokalen Verzeichnisses in einen Amazon Simple Storage Service (Amazon S3)-Bucket

Das folgende Codebeispiel zeigt, wie Sie ein lokales Verzeichnis rekursiv in einen Amazon Simple Storage Service (Amazon S3)-Bucket hochladen.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Verwenden Sie ein [S3 TransferManager](#), um [ein lokales Verzeichnis hochzuladen](#). Sehen Sie sich die [vollständige Datei](#) an und [testen](#) Sie sie.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.DirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.UploadDirectoryRequest;

import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

    public Integer uploadDirectory(S3TransferManager transferManager,
        URI sourceDirectory, String bucketName) {
        DirectoryUpload directoryUpload =
transferManager.uploadDirectory(UploadDirectoryRequest.builder()
            .source(Paths.get(sourceDirectory))
            .bucket(bucketName)
            .build());
```

```
CompletedDirectoryUpload completedDirectoryUpload =
directoryUpload.completionFuture().join();
    completedDirectoryUpload.failedTransfers()
        .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
    return completedDirectoryUpload.failedTransfers().size();
}
```

- API-Einheiten finden Sie [UploadDirectory](#) unter AWS SDK for Java 2.x API-Referenz.

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Laden Sie große Dateien zu und von Amazon S3 hoch oder laden Sie sie herunter mit einem AWS SDK

Die folgenden Codebeispiele veranschaulichen, wie Sie große Dateien zu S3 hochladen bzw. von S3 herunterladen.

Weitere Informationen finden Sie unter [Hochladen eines Objekts mit Multipart-Upload](#).

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Rufen Sie Funktionen auf, die Dateien mithilfe von Amazon S3 zu und von einem S3-Bucket übertragen. `TransferUtility`.

```
global using System.Text;
global using Amazon.S3;
global using Amazon.S3.Model;
global using Amazon.S3.Transfer;
```

```
global using TransferUtilityBasics;

// This Amazon S3 client uses the default user credentials
// defined for this computer.
using Microsoft.Extensions.Configuration;

IAmazonS3 client = new AmazonS3Client();
var transferUtil = new TransferUtility(client);
IConfiguration _configuration;

_configuration = new ConfigurationBuilder()
    .SetBasePath(Directory.GetCurrentDirectory())
    .AddJsonFile("settings.json") // Load test settings from JSON file.
    .AddJsonFile("settings.local.json",
        true) // Optionally load local settings.
    .Build();

// Edit the values in settings.json to use an S3 bucket and files that
// exist on your AWS account and on the local computer where you
// run this scenario.
var bucketName = _configuration["BucketName"];
var localPath =
    $"{Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData)}\
\TransferFolder";

DisplayInstructions();

PressEnter();

Console.WriteLine();

// Upload a single file to an S3 bucket.
DisplayTitle("Upload a single file");

var fileToUpload = _configuration["FileToUpload"];
Console.WriteLine($"Uploading {fileToUpload} to the S3 bucket, {bucketName}.");

var success = await TransferMethods.UploadSingleFileAsync(transferUtil,
    bucketName, fileToUpload, localPath);
if (success)
{
```

```
    Console.WriteLine($"Successfully uploaded the file, {fileToUpload} to
    {bucketName}.");
}

PressEnter();

// Upload a local directory to an S3 bucket.
DisplayTitle("Upload all files from a local directory");
Console.WriteLine("Upload all the files in a local folder to an S3 bucket.");
const string keyPrefix = "UploadFolder";
var uploadPath = $"{localPath}\\UploadFolder";

Console.WriteLine($"Uploading the files in {uploadPath} to {bucketName}");
DisplayTitle($"{uploadPath} files");
DisplayLocalFiles(uploadPath);
Console.WriteLine();

PressEnter();

success = await TransferMethods.UploadFullDirectoryAsync(transferUtil,
    bucketName, keyPrefix, uploadPath);
if (success)
{
    Console.WriteLine($"Successfully uploaded the files in {uploadPath} to
    {bucketName}.");
    Console.WriteLine($"{bucketName} currently contains the following files:");
    await DisplayBucketFiles(client, bucketName, keyPrefix);
    Console.WriteLine();
}

PressEnter();

// Download a single file from an S3 bucket.
DisplayTitle("Download a single file");
Console.WriteLine("Now we will download a single file from an S3 bucket.");

var keyName = _configuration["FileToDownload"];

Console.WriteLine($"Downloading {keyName} from {bucketName}.");

success = await TransferMethods.DownloadSingleFileAsync(transferUtil, bucketName,
    keyName, localPath);
if (success)
{
```

```
        Console.WriteLine($"Successfully downloaded the file, {keyName} from
        {bucketName}.");
    }

    PressEnter();

    // Download the contents of a directory from an S3 bucket.
    DisplayTitle("Download the contents of an S3 bucket");
    var s3Path = _configuration["S3Path"];
    var downloadPath = $"{localPath}\\{s3Path}";

    Console.WriteLine($"Downloading the contents of {bucketName}\\{s3Path}");
    Console.WriteLine($"{bucketName}\\{s3Path} contains the following files:");
    await DisplayBucketFiles(client, bucketName, s3Path);
    Console.WriteLine();

    success = await TransferMethods.DownloadS3DirectoryAsync(transferUtil,
        bucketName, s3Path, downloadPath);
    if (success)
    {
        Console.WriteLine($"Downloaded the files in {bucketName} to
        {downloadPath}.");
        Console.WriteLine($"{downloadPath} now contains the following files:");
        DisplayLocalFiles(downloadPath);
    }

    Console.WriteLine("\n\nThe TransferUtility Basics application has completed.");
    PressEnter();

    // Displays the title for a section of the scenario.
    static void DisplayTitle(string titleText)
    {
        var sepBar = new string('-', Console.WindowWidth);

        Console.WriteLine(sepBar);
        Console.WriteLine(CenterText(titleText));
        Console.WriteLine(sepBar);
    }

    // Displays a description of the actions to be performed by the scenario.
    static void DisplayInstructions()
    {
        var sepBar = new string('-', Console.WindowWidth);
```

```
    DisplayTitle("Amazon S3 Transfer Utility Basics");
    Console.WriteLine("This program shows how to use the Amazon S3 Transfer
Utility.");
    Console.WriteLine("It performs the following actions:");
    Console.WriteLine("\t1. Upload a single object to an S3 bucket.");
    Console.WriteLine("\t2. Upload an entire directory from the local computer to
an\n\t S3 bucket.");
    Console.WriteLine("\t3. Download a single object from an S3 bucket.");
    Console.WriteLine("\t4. Download the objects in an S3 bucket to a local
directory.");
    Console.WriteLine($" \n{sepBar}");
}

// Pauses the scenario.
static void PressEnter()
{
    Console.WriteLine("Press <Enter> to continue.");
    _ = Console.ReadLine();
    Console.WriteLine("\n");
}

// Returns the string textToCenter, padded on the left with spaces
// that center the text on the console display.
static string CenterText(string textToCenter)
{
    var centeredText = new StringBuilder();
    var screenWidth = Console.WindowWidth;
    centeredText.Append(new string(' ', (int)(screenWidth -
textToCenter.Length) / 2));
    centeredText.Append(textToCenter);
    return centeredText.ToString();
}

// Displays a list of file names included in the specified path.
static void DisplayLocalFiles(string localPath)
{
    var fileList = Directory.GetFiles(localPath);
    if (fileList.Length > 0)
    {
        foreach (var fileName in fileList)
        {
            Console.WriteLine(fileName);
        }
    }
}
```

```
}

// Displays a list of the files in the specified S3 bucket and prefix.
static async Task DisplayBucketFiles(IAmazonS3 client, string bucketName, string
    s3Path)
{
    ListObjectsV2Request request = new()
    {
        BucketName = bucketName,
        Prefix = s3Path,
        MaxKeys = 5,
    };

    var response = new ListObjectsV2Response();

    do
    {
        response = await client.ListObjectsV2Async(request);

        response.S3Objects
            .ForEach(obj => Console.WriteLine($"{obj.Key}"));

        // If the response is truncated, set the request ContinuationToken
        // from the NextContinuationToken property of the response.
        request.ContinuationToken = response.NextContinuationToken;
    } while (response.IsTruncated);
}
```

Laden Sie eine einzelne Datei hoch.

```
/// <summary>
/// Uploads a single file from the local computer to an S3 bucket.
/// </summary>
/// <param name="transferUtil">The transfer initialized TransferUtility
/// object.</param>
/// <param name="bucketName">The name of the S3 bucket where the file
/// will be stored.</param>
/// <param name="fileName">The name of the file to upload.</param>
/// <param name="localPath">The local path where the file is stored.</
param>
```

```
    /// <returns>A boolean value indicating the success of the action.</
returns>
    public static async Task<bool> UploadSingleFileAsync(
        TransferUtility transferUtil,
        string bucketName,
        string fileName,
        string localPath)
    {
        if (File.Exists($"{localPath}\\{fileName}"))
        {
            try
            {
                await transferUtil.UploadAsync(new
TransferUtilityUploadRequest
                {
                    BucketName = bucketName,
                    Key = fileName,
                    FilePath = $"{localPath}\\{fileName}",
                });

                return true;
            }
            catch (AmazonS3Exception s3Ex)
            {
                Console.WriteLine($"Could not upload {fileName} from
{localPath} because:");
                Console.WriteLine(s3Ex.Message);
                return false;
            }
        }
        else
        {
            Console.WriteLine($"{fileName} does not exist in {localPath}");
            return false;
        }
    }
}
```

Laden Sie ein ganzes lokales Verzeichnis hoch.

```
    /// <summary>
    /// Uploads all the files in a local directory to a directory in an S3
```



```
    /// bucket.
    /// </summary>
    /// <param name="transferUtil">The transfer initialized TransferUtility
    /// object.</param>
    /// <param name="bucketName">The name of the S3 bucket where the files
    /// will be stored.</param>
    /// <param name="keyPrefix">The key prefix is the S3 directory where
    /// the files will be stored.</param>
    /// <param name="localPath">The local directory that contains the files
    /// to be uploaded.</param>
    /// <returns>A Boolean value representing the success of the action.</
returns>
    public static async Task<bool> UploadFullDirectoryAsync(
        TransferUtility transferUtil,
        string bucketName,
        string keyPrefix,
        string localPath)
    {
        if (Directory.Exists(localPath))
        {
            try
            {
                await transferUtil.UploadDirectoryAsync(new
TransferUtilityUploadDirectoryRequest
                {
                    BucketName = bucketName,
                    KeyPrefix = keyPrefix,
                    Directory = localPath,
                });

                return true;
            }
            catch (AmazonS3Exception s3Ex)
            {
                Console.WriteLine($"Can't upload the contents of {localPath}
because:");

                Console.WriteLine(s3Ex?.Message);
                return false;
            }
        }
        else
        {
            Console.WriteLine($"The directory {localPath} does not exist.");
            return false;
        }
    }
}
```

```
    }
}
```

Laden Sie eine einzelne Datei herunter.

```
/// <summary>
/// Download a single file from an S3 bucket to the local computer.
/// </summary>
/// <param name="transferUtil">The transfer initialized TransferUtility
/// object.</param>
/// <param name="bucketName">The name of the S3 bucket containing the
/// file to download.</param>
/// <param name="keyName">The name of the file to download.</param>
/// <param name="localPath">The path on the local computer where the
/// downloaded file will be saved.</param>
/// <returns>A Boolean value indicating the results of the action.</
returns>
public static async Task<bool> DownloadSingleFileAsync(
    TransferUtility transferUtil,
    string bucketName,
    string keyName,
    string localPath)
{
    await transferUtil.DownloadAsync(new TransferUtilityDownloadRequest
    {
        BucketName = bucketName,
        Key = keyName,
        FilePath = $"{localPath}\\{keyName}",
    });

    return (File.Exists($"{localPath}\\{keyName}"));
}
```

Laden Sie den Inhalt eines S3 Buckets herunter.

```
/// <summary>
/// Downloads the contents of a directory in an S3 bucket to a
```

```
    /// directory on the local computer.
    /// </summary>
    /// <param name="transferUtil">The transfer initialized TransferUtility
    /// object.</param>
    /// <param name="bucketName">The bucket containing the files to
download.</param>
    /// <param name="s3Path">The S3 directory where the files are located.</
param>
    /// <param name="localPath">The local path to which the files will be
    /// saved.</param>
    /// <returns>A Boolean value representing the success of the action.</
returns>
    public static async Task<bool> DownloadS3DirectoryAsync(
        TransferUtility transferUtil,
        string bucketName,
        string s3Path,
        string localPath)
    {
        int fileCount = 0;

        // If the directory doesn't exist, it will be created.
        if (Directory.Exists(s3Path))
        {
            var files = Directory.GetFiles(localPath);
            fileCount = files.Length;
        }

        await transferUtil.DownloadDirectoryAsync(new
TransferUtilityDownloadDirectoryRequest
        {
            BucketName = bucketName,
            LocalDirectory = localPath,
            S3Directory = s3Path,
        });

        if (Directory.Exists(localPath))
        {
            var files = Directory.GetFiles(localPath);
            if (files.Length > fileCount)
            {
                return true;
            }
        }

        // No change in the number of files. Assume
```

```
        // the download failed.
        return false;
    }

    // The local directory doesn't exist. No files
    // were downloaded.
    return false;
}
```

Verfolgen Sie den Fortschritt eines Uploads mit dem TransferUtility.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Transfer;

/// <summary>
/// This example shows how to track the progress of a multipart upload
/// using the Amazon Simple Storage Service (Amazon S3) TransferUtility to
/// upload to an Amazon S3 bucket.
/// </summary>
public class TrackMPUUsingHighLevelAPI
{
    public static async Task Main()
    {
        string bucketName = "amzn-s3-demo-bucket";
        string keyName = "sample_pic.png";
        string path = "filepath/directory/";
        string filePath = $"{path}{keyName}";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
        IAmazonS3 client = new AmazonS3Client();

        await TrackMPUAsync(client, bucketName, filePath, keyName);
    }

    /// <summary>
    /// Starts an Amazon S3 multipart upload and assigns an event handler to
```

```
/// track the progress of the upload.
/// </summary>
/// <param name="client">The initialized Amazon S3 client object used to
/// perform the multipart upload.</param>
/// <param name="bucketName">The name of the bucket to which to upload
/// the file.</param>
/// <param name="filePath">The path, including the file name of the
/// file to be uploaded to the Amazon S3 bucket.</param>
/// <param name="keyName">The file name to be used in the
/// destination Amazon S3 bucket.</param>
public static async Task TrackMPUAsync(
    IAmazonS3 client,
    string bucketName,
    string filePath,
    string keyName)
{
    try
    {
        var fileTransferUtility = new TransferUtility(client);

        // Use TransferUtilityUploadRequest to configure options.
        // In this example we subscribe to an event.
        var uploadRequest =
            new TransferUtilityUploadRequest
            {
                BucketName = bucketName,
                FilePath = filePath,
                Key = keyName,
            };

        uploadRequest.UploadProgressEvent +=
            new EventHandler<UploadProgressArgs>(
                UploadRequest_UploadPartProgressEvent);

        await fileTransferUtility.UploadAsync(uploadRequest);
        Console.WriteLine("Upload completed");
    }
    catch (AmazonS3Exception ex)
    {
        Console.WriteLine($"Error:: {ex.Message}");
    }
}

/// <summary>
```

```
/// Event handler to check the progress of the multipart upload.
/// </summary>
/// <param name="sender">The object that raised the event.</param>
/// <param name="e">The object that contains multipart upload
/// information.</param>
public static void UploadRequest_UploadPartProgressEvent(object sender,
UploadProgressArgs e)
{
    // Process event.
    Console.WriteLine($"{e.TransferredBytes}/{e.TotalBytes}");
}
}
```

Laden Sie ein Objekt mit Verschlüsselung hoch.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// Uses the Amazon Simple Storage Service (Amazon S3) low level API to
/// perform a multipart upload to an Amazon S3 bucket.
/// </summary>
public class SSECLowLevelMPUCopyObject
{
    public static async Task Main()
    {
        string existingBucketName = "amzn-s3-demo-bucket";
        string sourceKeyName = "sample_file.txt";
        string targetKeyName = "sample_file_copy.txt";
        string filePath = $"sample\\{targetKeyName}";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USEast1.
        IAmazonS3 client = new AmazonS3Client();
```

```
// Create the encryption key.
var base64Key = CreateEncryptionKey();

await CreateSampleObjUsingClientEncryptionKeyAsync(
    client,
    existingBucketName,
    sourceKeyName,
    filePath,
    base64Key);
}

/// <summary>
/// Creates the encryption key to use with the multipart upload.
/// </summary>
/// <returns>A string containing the base64-encoded key for encrypting
/// the multipart upload.</returns>
public static string CreateEncryptionKey()
{
    Aes aesEncryption = Aes.Create();
    aesEncryption.KeySize = 256;
    aesEncryption.GenerateKey();
    string base64Key = Convert.ToBase64String(aesEncryption.Key);
    return base64Key;
}

/// <summary>
/// Creates and uploads an object using a multipart upload.
/// </summary>
/// <param name="client">The initialized Amazon S3 object used to
/// initialize and perform the multipart upload.</param>
/// <param name="existingBucketName">The name of the bucket to which
/// the object will be uploaded.</param>
/// <param name="sourceKeyName">The source object name.</param>
/// <param name="filePath">The location of the source object.</param>
/// <param name="base64Key">The encryption key to use with the upload.</
param>
public static async Task CreateSampleObjUsingClientEncryptionKeyAsync(
    IAmazonS3 client,
    string existingBucketName,
    string sourceKeyName,
    string filePath,
    string base64Key)
{
```

```
List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = sourceKeyName,
    ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key,
};

InitiateMultipartUploadResponse initResponse =
    await client.InitiateMultipartUploadAsync(initiateRequest);

long contentLength = new FileInfo(filePath).Length;
long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

try
{
    long filePosition = 0;
    for (int i = 1; filePosition < contentLength; i++)
    {
        UploadPartRequest uploadRequest = new UploadPartRequest
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
            UploadId = initResponse.UploadId,
            PartNumber = i,
            PartSize = partSize,
            FilePosition = filePosition,
            FilePath = filePath,
            ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
            ServerSideEncryptionCustomerProvidedKey = base64Key,
        };

        // Upload part and add response to our list.
        uploadResponses.Add(await
client.UploadPartAsync(uploadRequest));

        filePosition += partSize;
    }
}
```



```
        CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
    {
        BucketName = existingBucketName,
        Key = sourceKeyName,
        UploadId = initResponse.UploadId,
    };
    completeRequest.AddPartETags(uploadResponses);


    CompleteMultipartUploadResponse completeUploadResponse =
        await client.CompleteMultipartUploadAsync(completeRequest);
    }
    catch (Exception exception)
    {
        Console.WriteLine($"Exception occurred: {exception.Message}");

        // If there was an error, abort the multipart upload.
        AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
    {
        BucketName = existingBucketName,
        Key = sourceKeyName,
        UploadId = initResponse.UploadId,
    };

        await client.AbortMultipartUploadAsync(abortMPURequest);
    }
    }
}
```

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Laden Sie ein großes Objekt hoch, indem Sie einen Upload-Manager verwenden, um die Daten in Teile zu zerlegen und sie gleichzeitig hochzuladen.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
// actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
// bucket
// and object actions.
type BucketBasics struct {
    S3Client *s3.Client
}

// UploadLargeObject uses an upload manager to upload data to an object in a
// bucket.
// The upload manager breaks large data into parts and uploads the parts
// concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
    largeObject []byte) error {
    largeBuffer := bytes.NewReader(largeObject)
    var partMiBs int64 = 10
    uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
        u.PartSize = partMiBs * 1024 * 1024
    })
    _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
        Bucket: aws.String(bucketName),
        Key:     aws.String(objectKey),
        Body:   largeBuffer,
    })
    if err != nil {
        log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }

    return err
}
```

Laden Sie ein großes Objekt herunter, indem Sie einen Download-Manager verwenden, um die Daten in Teilen abzurufen und sie gleichzeitig herunterzuladen.

```
// DownloadLargeObject uses a download manager to download an object from a
// bucket.
// The download manager gets the data in parts and writes them to a buffer until
// all of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey
string) ([]byte, error) {
    var partMiBs int64 = 10
    downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader)
    {
        d.PartSize = partMiBs * 1024 * 1024
    })
    buffer := manager.NewWriteAtBuffer([]byte{})
    _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
        Bucket: aws.String(bucketName),
        Key:     aws.String(objectKey),
    })
    if err != nil {
        log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
            bucketName, objectKey, err)
    }
    return buffer.Bytes(), err
}
```

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Rufen Sie Funktionen auf, die Dateien mithilfe von S3 zu und von einem S3-Bucket übertragen.
`TransferManager`.

```
public Integer downloadObjectsToDirectory(S3TransferManager transferManager,
    URI destinationPathURI, String bucketName) {
    DirectoryDownload directoryDownload =
transferManager.downloadDirectory(DownloadDirectoryRequest.builder()
    .destination(Paths.get(destinationPathURI))
    .bucket(bucketName)
    .build());
    CompletedDirectoryDownload completedDirectoryDownload =
directoryDownload.completionFuture().join();

    completedDirectoryDownload.failedTransfers()
        .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
    return completedDirectoryDownload.failedTransfers().size();
}
```

Laden Sie ein ganzes lokales Verzeichnis hoch.

```
public Integer uploadDirectory(S3TransferManager transferManager,
    URI sourceDirectory, String bucketName) {
    DirectoryUpload directoryUpload =
transferManager.uploadDirectory(UploadDirectoryRequest.builder()
    .source(Paths.get(sourceDirectory))
    .bucket(bucketName)
    .build());

    CompletedDirectoryUpload completedDirectoryUpload =
directoryUpload.completionFuture().join();
    completedDirectoryUpload.failedTransfers()
        .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
    return completedDirectoryUpload.failedTransfers().size();
}
```

Laden Sie eine einzelne Datei hoch.

```
public String uploadFile(S3TransferManager transferManager, String
bucketName,
                        String key, URI filePathURI) {
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b.bucket(bucketName).key(key))
        .source(Paths.get(filePathURI))
        .build();

    FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);

    CompletedFileUpload uploadResult = fileUpload.completionFuture().join();
    return uploadResult.response().eTag();
}
```

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Laden Sie eine große Datei hoch.

```
import {
    CreateMultipartUploadCommand,
    UploadPartCommand,
    CompleteMultipartUploadCommand,
    AbortMultipartUploadCommand,
    S3Client,
} from "@aws-sdk/client-s3";

const twentyFiveMB = 25 * 1024 * 1024;

export const createString = (size = twentyFiveMB) => {
    return "x".repeat(size);
};
```

```
export const main = async () => {
  const s3Client = new S3Client({});
  const bucketName = "test-bucket";
  const key = "multipart.txt";
  const str = createString();
  const buffer = Buffer.from(str, "utf8");

  let uploadId;

  try {
    const multipartUpload = await s3Client.send(
      new CreateMultipartUploadCommand({
        Bucket: bucketName,
        Key: key,
      }),
    );

    uploadId = multipartUpload.UploadId;

    const uploadPromises = [];
    // Multipart uploads require a minimum size of 5 MB per part.
    const partSize = Math.ceil(buffer.length / 5);

    // Upload each part.
    for (let i = 0; i < 5; i++) {
      const start = i * partSize;
      const end = start + partSize;
      uploadPromises.push(
        s3Client
          .send(
            new UploadPartCommand({
              Bucket: bucketName,
              Key: key,
              UploadId: uploadId,
              Body: buffer.subarray(start, end),
              PartNumber: i + 1,
            }),
          )
          .then((d) => {
            console.log("Part", i + 1, "uploaded");
            return d;
          }),
      );
    }
  }
}
```

```
const uploadResults = await Promise.all(uploadPromises);

return await s3Client.send(
  new CompleteMultipartUploadCommand({
    Bucket: bucketName,
    Key: key,
    UploadId: uploadId,
    MultipartUpload: {
      Parts: uploadResults.map(({ ETag }, i) => ({
        ETag,
        PartNumber: i + 1,
      })),
    },
  }),
);

// Verify the output by downloading the file from the Amazon Simple Storage
// Service (Amazon S3) console.
// Because the output is a 25 MB string, text editors might struggle to open
// the file.
} catch (err) {
  console.error(err);

  if (uploadId) {
    const abortCommand = new AbortMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
      UploadId: uploadId,
    });

    await s3Client.send(abortCommand);
  }
}
};
```

Laden Sie eine große Datei herunter.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream } from "fs";

const s3Client = new S3Client({});
```

```
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
    Bucket: bucket,
    Key: key,
    Range: `bytes=${start}-${end}`,
  });

  return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
  const [start, end] = range.split("-");
  return {
    start: parseInt(start),
    end: parseInt(end),
    length: parseInt(length),
  };
};

export const isComplete = ({ end, length }) => end === length - 1;

// When downloading a large file, you might want to break it down into
// smaller pieces. Amazon S3 accepts a Range header to specify the start
// and end of the byte range to be downloaded.
const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    console.log(`Downloading bytes ${nextRange.start} to ${nextRange.end}`);

    const { ContentRange, Body } = await getObjectRange({
```



```
    bucket,
    key,
    ...nextRange,
  });

  writeStream.write(await Body.transformToByteArray());
  rangeAndLength = getRangeAndLength(ContentRange);
}
};

export const main = async () => {
  await downloadInChunks({
    bucket: "my-cool-bucket",
    key: "my-cool-object.txt",
  });
};
```

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie Funktionen, die Dateien mit mehreren der verfügbaren Transfer-Manager-Einstellungen übertragen. Verwenden Sie eine Callback-Klasse, um während der Dateiübertragung Callback-Fortschritte zu schreiben.

```
import sys
import threading

import boto3
from boto3.s3.transfer import TransferConfig

MB = 1024 * 1024
s3 = boto3.resource("s3")
```

```
class TransferCallback:
    """
    Handle callbacks from the transfer manager.

    The transfer manager periodically calls the __call__ method throughout
    the upload and download process so that it can take action, such as
    displaying progress to the user and collecting data about the transfer.
    """

    def __init__(self, target_size):
        self._target_size = target_size
        self._total_transferred = 0
        self._lock = threading.Lock()
        self.thread_info = {}

    def __call__(self, bytes_transferred):
        """
        The callback method that is called by the transfer manager.

        Display progress during file transfer and collect per-thread transfer
        data. This method can be called by multiple threads, so shared instance
        data is protected by a thread lock.
        """
        thread = threading.current_thread()
        with self._lock:
            self._total_transferred += bytes_transferred
            if thread.ident not in self.thread_info.keys():
                self.thread_info[thread.ident] = bytes_transferred
            else:
                self.thread_info[thread.ident] += bytes_transferred

            target = self._target_size * MB
            sys.stdout.write(
                f"\r{self._total_transferred} of {target} transferred "
                f"({(self._total_transferred / target) * 100:.2f}%)."
            )
            sys.stdout.flush()

    def upload_with_default_configuration(
        local_file_path, bucket_name, object_key, file_size_mb
    ):

```

```
"""
Upload a file from a local folder to an Amazon S3 bucket, using the default
configuration.
"""
transfer_callback = TransferCallback(file_size_mb)
s3.Bucket(bucket_name).upload_file(
    local_file_path, object_key, Callback=transfer_callback
)
return transfer_callback.thread_info

def upload_with_chunksize_and_meta(
    local_file_path, bucket_name, object_key, file_size_mb, metadata=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {"Metadata": metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
        Callback=transfer_callback,
    )
    return transfer_callback.thread_info

def upload_with_high_threshold(local_file_path, bucket_name, object_key,
    file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.
```

Setting a multipart threshold larger than the size of the file results in the transfer manager sending the file as a standard upload instead of a multipart upload.

```
"""
transfer_callback = TransferCallback(file_size_mb)
config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
s3.Bucket(bucket_name).upload_file(
    local_file_path, object_key, Config=config, Callback=transfer_callback
)
return transfer_callback.thread_info

def upload_with_sse(
    local_file_path, bucket_name, object_key, file_size_mb, sse_key=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey":
sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, ExtraArgs=extra_args,
Callback=transfer_callback
    )
    return transfer_callback.thread_info

def download_with_default_configuration(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using the
    default configuration.
    """
```

```
transfer_callback = TransferCallback(file_size_mb)
s3.Bucket(bucket_name).Object(object_key).download_file(
    download_file_path, Callback=transfer_callback
)
return transfer_callback.thread_info

def download_with_single_thread(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using a
    single thread.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(use_threads=False)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info

def download_with_high_threshold(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard download instead
    of a multipart download.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info

def download_with_sse(
    bucket_name, object_key, download_file_path, file_size_mb, sse_key
):
```

```

"""
Download a file from an Amazon S3 bucket to a local folder, adding a
customer-provided encryption key to the request.

When this kind of encryption is specified, Amazon S3 encrypts the object
at rest and allows downloads only when the expected encryption key is
provided in the download request.
"""
transfer_callback = TransferCallback(file_size_mb)

if sse_key:
    extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey":
sse_key}
else:
    extra_args = None
s3.Bucket(bucket_name).Object(object_key).download_file(
    download_file_path, ExtraArgs=extra_args, Callback=transfer_callback
)
return transfer_callback.thread_info

```

Demonstrieren Sie die Funktionen des Transfer-Managers und melden Sie Ergebnisse.

```

import hashlib
import os
import platform
import shutil
import time

import boto3
from boto3.s3.transfer import TransferConfig
from botocore.exceptions import ClientError
from botocore.exceptions import ParamValidationError
from botocore.exceptions import NoCredentialsError

import file_transfer

MB = 1024 * 1024
# These configuration attributes affect both uploads and downloads.
CONFIG_ATTRS = (
    "multipart_threshold",

```

```
    "multipart_chunksize",
    "max_concurrency",
    "use_threads",
)
# These configuration attributes affect only downloads.
DOWNLOAD_CONFIG_ATTRS = ("max_io_queue", "io_chunksize", "num_download_attempts")

class TransferDemoManager:
    """
    Manages the demonstration. Collects user input from a command line, reports
    transfer results, maintains a list of artifacts created during the
    demonstration, and cleans them up after the demonstration is completed.
    """

    def __init__(self):
        self._s3 = boto3.resource("s3")
        self._chore_list = []
        self._create_file_cmd = None
        self._size_multiplier = 0
        self.file_size_mb = 30
        self.demo_folder = None
        self.demo_bucket = None
        self._setup_platform_specific()
        self._terminal_width = shutil.get_terminal_size(fallback=(80, 80))[0]

    def collect_user_info(self):
        """
        Collect local folder and Amazon S3 bucket name from the user. These
        locations are used to store files during the demonstration.
        """
        while not self.demo_folder:
            self.demo_folder = input(
                "Which file folder do you want to use to store " "demonstration
files? "
            )
            if not os.path.isdir(self.demo_folder):
                print(f"{self.demo_folder} isn't a folder!")
                self.demo_folder = None

        while not self.demo_bucket:
            self.demo_bucket = input(
                "Which Amazon S3 bucket do you want to use to store "
"demonstration files? "
            )
```

```
)
try:
    self._s3.meta.client.head_bucket(Bucket=self.demo_bucket)
except ParamValidationError as err:
    print(err)
    self.demo_bucket = None
except ClientError as err:
    print(err)
    print(
        f"Either {self.demo_bucket} doesn't exist or you don't "
        f"have access to it."
    )
    self.demo_bucket = None

def demo(
    self, question, upload_func, download_func, upload_args=None,
download_args=None
):
    """Run a demonstration.

    Ask the user if they want to run this specific demonstration.
    If they say yes, create a file on the local path, upload it
    using the specified upload function, then download it using the
    specified download function.
    """
    if download_args is None:
        download_args = {}
    if upload_args is None:
        upload_args = {}
    question = question.format(self.file_size_mb)
    answer = input(f"{question} (y/n)")
    if answer.lower() == "y":
        local_file_path, object_key, download_file_path =
self._create_demo_file()

        file_transfer.TransferConfig = self._config_wrapper(
            TransferConfig, CONFIG_ATTRS
        )
        self._report_transfer_params(
            "Uploading", local_file_path, object_key, **upload_args
        )
        start_time = time.perf_counter()
        thread_info = upload_func(
            local_file_path,
```



```
        self.demo_bucket,
        object_key,
        self.file_size_mb,
        **upload_args,
    )
    end_time = time.perf_counter()
    self._report_transfer_result(thread_info, end_time - start_time)

    file_transfer.TransferConfig = self._config_wrapper(
        TransferConfig, CONFIG_ATTRS + DOWNLOAD_CONFIG_ATTRS
    )
    self._report_transfer_params(
        "Downloading", object_key, download_file_path, **download_args
    )
    start_time = time.perf_counter()
    thread_info = download_func(
        self.demo_bucket,
        object_key,
        download_file_path,
        self.file_size_mb,
        **download_args,
    )
    end_time = time.perf_counter()
    self._report_transfer_result(thread_info, end_time - start_time)

def last_name_set(self):
    """Get the name set used for the last demo."""
    return self._chore_list[-1]

def cleanup(self):
    """
    Remove files from the demo folder, and uploaded objects from the
    Amazon S3 bucket.
    """
    print("-" * self._terminal_width)
    for local_file_path, s3_object_key, downloaded_file_path in
self._chore_list:
        print(f"Removing {local_file_path}")
        try:
            os.remove(local_file_path)
        except FileNotFoundError as err:
            print(err)

        print(f"Removing {downloaded_file_path}")
```

```
        try:
            os.remove(downloaded_file_path)
        except FileNotFoundError as err:
            print(err)

        if self.demo_bucket:
            print(f"Removing {self.demo_bucket}:{s3_object_key}")
            try:
                self._s3.Bucket(self.demo_bucket).Object(s3_object_key).delete()
            except ClientError as err:
                print(err)

    def _setup_platform_specific(self):
        """Set up platform-specific command used to create a large file."""
        if platform.system() == "Windows":
            self._create_file_cmd = "fsutil file createnew {} {}"
            self._size_multiplier = MB
        elif platform.system() == "Linux" or platform.system() == "Darwin":
            self._create_file_cmd = f"dd if=/dev/urandom of={{}} " f"bs={{MB}}
count={{}}"
            self._size_multiplier = 1
        else:
            raise EnvironmentError(
                f"Demo of platform {platform.system()} isn't supported."
            )

    def _create_demo_file(self):
        """
        Create a file in the demo folder specified by the user. Store the local
        path, object name, and download path for later cleanup.

        Only the local file is created by this method. The Amazon S3 object and
        download file are created later during the demonstration.

        Returns:
        A tuple that contains the local file path, object name, and download
        file path.
        """
        file_name_template = "TestFile{}-{}.demo"
        local_suffix = "local"
        object_suffix = "s3object"
        download_suffix = "downloaded"
        file_tag = len(self._chore_list) + 1
```

```
local_file_path = os.path.join(
    self.demo_folder, file_name_template.format(file_tag, local_suffix)
)

s3_object_key = file_name_template.format(file_tag, object_suffix)

downloaded_file_path = os.path.join(
    self.demo_folder, file_name_template.format(file_tag,
download_suffix)
)

filled_cmd = self._create_file_cmd.format(
    local_file_path, self.file_size_mb * self._size_multiplier
)

print(
    f"Creating file of size {self.file_size_mb} MB "
    f"in {self.demo_folder} by running:"
)
print(f"{'':4}{filled_cmd}")
os.system(filled_cmd)

chore = (local_file_path, s3_object_key, downloaded_file_path)
self._chore_list.append(chore)
return chore

def _report_transfer_params(self, verb, source_name, dest_name, **kwargs):
    """Report configuration and extra arguments used for a file transfer."""
    print("-" * self._terminal_width)
    print(f"{verb} {source_name} ({self.file_size_mb} MB) to {dest_name}")
    if kwargs:
        print("With extra args:")
        for arg, value in kwargs.items():
            print(f"{'':4}{arg:<20}: {value}')

    @staticmethod
    def ask_user(question):
        """
        Ask the user a yes or no question.

        Returns:
        True when the user answers 'y' or 'Y'; otherwise, False.
        """
```

```

        answer = input(f"{question} (y/n) ")
        return answer.lower() == "y"

    @staticmethod
    def _config_wrapper(func, config_attrs):
        def wrapper(*args, **kwargs):
            config = func(*args, **kwargs)
            print("With configuration:")
            for attr in config_attrs:
                print(f'{"":4}{attr:<20}: {getattr(config, attr)}')
            return config

        return wrapper

    @staticmethod
    def _report_transfer_result(thread_info, elapsed):
        """Report the result of a transfer, including per-thread data."""
        print(f"\nUsed {len(thread_info)} threads.")
        for ident, byte_count in thread_info.items():
            print(f'{"":4}Thread {ident} copied {byte_count} bytes.")
        print(f"Your transfer took {elapsed:.2f} seconds.")

def main():
    """
    Run the demonstration script for s3_file_transfer.
    """
    demo_manager = TransferDemoManager()
    demo_manager.collect_user_info()

    # Upload and download with default configuration. Because the file is 30 MB
    # and the default multipart_threshold is 8 MB, both upload and download are
    # multipart transfers.
    demo_manager.demo(
        "Do you want to upload and download a {} MB file "
        "using the default configuration?",
        file_transfer.upload_with_default_configuration,
        file_transfer.download_with_default_configuration,
    )

    # Upload and download with multipart_threshold set higher than the size of
    # the file. This causes the transfer manager to use standard transfers
    # instead of multipart transfers.
    demo_manager.demo(

```

```
    "Do you want to upload and download a {} MB file "  
    "as a standard (not multipart) transfer?",  
    file_transfer.upload_with_high_threshold,  
    file_transfer.download_with_high_threshold,  
    )  
  
# Upload with specific chunk size and additional metadata.  
# Download with a single thread.  
demo_manager.demo(  
    "Do you want to upload a {} MB file with a smaller chunk size and "  
    "then download the same file using a single thread?",  
    file_transfer.upload_with_chunksize_and_meta,  
    file_transfer.download_with_single_thread,  
    upload_args={  
        "metadata": {  
            "upload_type": "chunky",  
            "favorite_color": "aqua",  
            "size": "medium",  
        }  
    },  
    )  
  
# Upload using server-side encryption with customer-provided  
# encryption keys.  
# Generate a 256-bit key from a passphrase.  
sse_key = hashlib.sha256("demo_passphrase".encode("utf-8")).digest()  
demo_manager.demo(  
    "Do you want to upload and download a {} MB file using "  
    "server-side encryption?",  
    file_transfer.upload_with_sse,  
    file_transfer.download_with_sse,  
    upload_args={"sse_key": sse_key},  
    download_args={"sse_key": sse_key},  
    )  
  
# Download without specifying an encryption key to show that the  
# encryption key must be included to download an encrypted object.  
if demo_manager.ask_user(  
    "Do you want to try to download the encrypted "  
    "object without sending the required key?"  
):  
    try:  
        _, object_key, download_file_path = demo_manager.last_name_set()  
        file_transfer.download_with_default_configuration(  

```

```
        demo_manager.demo_bucket,
        object_key,
        download_file_path,
        demo_manager.file_size_mb,
    )
except ClientError as err:
    print(
        "Got expected error when trying to download an encrypted "
        "object without specifying encryption info:"
    )
    print(f"{'':4}{err}")

# Remove all created and downloaded files, remove all objects from
# S3 storage.
if demo_manager.ask_user(
    "Demonstration complete. Do you want to remove local files " "and S3
objects?"
):
    demo_manager.cleanup()

if __name__ == "__main__":
    try:
        main()
    except NoCredentialsError as error:
        print(error)
        print(
            "To run this example, you must have valid credentials in "
            "a shared credential file or set in environment variables."
        )
```

Rust

SDK für Rust

Note

Es gibt noch mehr dazu GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

```
use std::fs::File;
use std::io::prelude::*;
use std::path::Path;

use aws_config::meta::region::RegionProviderChain;
use aws_sdk_s3::error::DisplayErrorContext;
use aws_sdk_s3::operation::{
    create_multipart_upload::CreateMultipartUploadOutput,
    get_object::GetObjectOutput,
};
use aws_sdk_s3::types::{CompletedMultipartUpload, CompletedPart};
use aws_sdk_s3::{config::Region, Client as S3Client};
use aws_smithy_types::byte_stream::{ByteStream, Length};
use rand::distributions::Alphanumeric;
use rand::{thread_rng, Rng};
use s3_code_examples::error::S3ExampleError;
use std::process;
use uuid::Uuid;

//In bytes, minimum chunk size of 5MB. Increase CHUNK_SIZE to send larger chunks.
const CHUNK_SIZE: u64 = 1024 * 1024 * 5;
const MAX_CHUNKS: u64 = 10000;

#[tokio::main]
pub async fn main() {
    if let Err(err) = run_example().await {
        eprintln!("Error: {}", DisplayErrorContext(err));
        process::exit(1);
    }
}

async fn run_example() -> Result<(), S3ExampleError> {
    let shared_config = aws_config::load_from_env().await;
    let client = S3Client::new(&shared_config);

    let bucket_name = format!("amzn-s3-demo-bucket-{}", Uuid::new_v4());
    let region_provider = RegionProviderChain::first_try(Region::new("us-west-2"));
    let region = region_provider.region().await.unwrap();
    s3_code_examples::create_bucket(&client, &bucket_name, &region).await?;

    let key = "sample.txt".to_string();
```

```
// Create a multipart upload. Use UploadPart and CompleteMultipartUpload to
// upload the file.
let multipart_upload_res: CreateMultipartUploadOutput = client
    .create_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .send()
    .await?;

let upload_id = multipart_upload_res.upload_id().ok_or(S3ExampleError::new(
    "Missing upload_id after CreateMultipartUpload",
))?;

//Create a file of random characters for the upload.
let mut file = File::create(&key).expect("Could not create sample file.");
// Loop until the file is 5 chunks.
while file.metadata().unwrap().len() <= CHUNK_SIZE * 4 {
    let rand_string: String = thread_rng()
        .sample_iter(&Alphanumeric)
        .take(256)
        .map(char::from)
        .collect();
    let return_string: String = "\n".to_string();
    file.write_all(rand_string.as_ref())
        .expect("Error writing to file.");
    file.write_all(return_string.as_ref())
        .expect("Error writing to file.");
}

let path = Path::new(&key);
let file_size = tokio::fs::metadata(path)
    .await
    .expect("it exists I swear")
    .len();

let mut chunk_count = (file_size / CHUNK_SIZE) + 1;
let mut size_of_last_chunk = file_size % CHUNK_SIZE;
if size_of_last_chunk == 0 {
    size_of_last_chunk = CHUNK_SIZE;
    chunk_count -= 1;
}

if file_size == 0 {
    return Err(S3ExampleError::new("Bad file size."));
```



```
}
if chunk_count > MAX_CHUNKS {
    return Err(S3ExampleError::new(
        "Too many chunks! Try increasing your chunk size.",
    ));
}

let mut upload_parts: Vec<aws_sdk_s3::types::CompletedPart> = Vec::new();

for chunk_index in 0..chunk_count {
    let this_chunk = if chunk_count - 1 == chunk_index {
        size_of_last_chunk
    } else {
        CHUNK_SIZE
    };
    let stream = ByteStream::read_from()
        .path(path)
        .offset(chunk_index * CHUNK_SIZE)
        .length(Length::Exact(this_chunk))
        .build()
        .await
        .unwrap();

    // Chunk index needs to start at 0, but part numbers start at 1.
    let part_number = (chunk_index as i32) + 1;
    let upload_part_res = client
        .upload_part()
        .key(&key)
        .bucket(&bucket_name)
        .upload_id(upload_id)
        .body(stream)
        .part_number(part_number)
        .send()
        .await?;

    upload_parts.push(
        CompletedPart::builder()
            .e_tag(upload_part_res.e_tag.unwrap_or_default())
            .part_number(part_number)
            .build(),
    );
}

// upload_parts: Vec<aws_sdk_s3::types::CompletedPart>
```

```
    let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
    .set_parts(Some(upload_parts))
    .build();

    let _complete_multipart_upload_res = client
    .complete_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .multipart_upload(completed_multipart_upload)
    .upload_id(upload_id)
    .send()
    .await?;

    let data: GetObjectOutput =
    s3_code_examples::download_object(&client, &bucket_name, &key).await?;
    let data_length: u64 = data
    .content_length()
    .unwrap_or_default()
    .try_into()
    .unwrap();
    if file.metadata().unwrap().len() == data_length {
        println!("Data lengths match.");
    } else {
        println!("The data was not the same size!");
    }

    s3_code_examples::clear_bucket(&client, &bucket_name)
    .await
    .expect("Error emptying bucket.");
    s3_code_examples::delete_bucket(&client, &bucket_name)
    .await
    .expect("Error deleting bucket.");

    Ok(())
}
```

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Laden Sie einen Stream unbekannter Größe mit einem in ein Amazon S3 S3-Objekt hoch AWS SDK

Im folgenden Codebeispiel wird veranschaulicht, wie Sie einen Stream unbekannter Größe in ein Amazon S3-Objekt hochladen.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Verwenden der [AWS CRTbasierter S3-Client](#).

```
import com.example.s3.util.AsyncExampleUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;

import java.io.ByteArrayInputStream;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

/**
 * @param s3CrtAsyncClient - To upload content from a stream of unknown
 * size, use the AWS CRT-based S3 client. For more information, see
 * https://docs.aws.amazon.com/sdk-for-java/latest/
 * developer-guide/crt-based-s3-client.html.
 * @param bucketName - The name of the bucket.
 * @param key - The name of the object.
 * @return software.amazon.awssdk.services.s3.model.PutObjectResponse -
 * Returns metadata pertaining to the put object operation.
 */
```

```
public PutObjectResponse putObjectFromStream(S3AsyncClient s3CrtAsyncClient,
String bucketName, String key) {

    BlockingInputStreamAsyncRequestBody body =
        AsyncRequestBody.forBlockingInputStream(null); // 'null'
    indicates a stream will be provided later.

    CompletableFuture<PutObjectResponse> responseFuture =
        s3CrtAsyncClient.putObject(r -> r.bucket(bucketName).key(key),
    body);

    // AsyncExampleUtils.randomString() returns a random string up to 100
    characters.
    String randomString = AsyncExampleUtils.randomString();
    logger.info("random string to upload: {}: length={}", randomString,
    randomString.length());

    // Provide the stream of data to be uploaded.
    body.writeInputStream(new ByteArrayInputStream(randomString.getBytes()));

    PutObjectResponse response = responseFuture.join(); // Wait for the
    response.
    logger.info("Object {} uploaded to bucket {}.", key, bucketName);
    return response;
}
}
```

Verwenden Sie den [Amazon-S3-Transfer-Manager](#).

```
import com.example.s3.util.AsyncExampleUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedUpload;
import software.amazon.awssdk.transfer.s3.model.Upload;

import java.io.ByteArrayInputStream;
import java.util.UUID;
```

```
/**
 * @param transferManager - To upload content from a stream of unknown size,
 use the S3TransferManager based on the AWS CRT-based S3 client.
 *
 *           For more information, see https://
 docs.aws.amazon.com/sdk-for-java/latest/developer-guide/transfer-manager.html.
 * @param bucketName - The name of the bucket.
 * @param key - The name of the object.
 * @return - software.amazon.awssdk.transfer.s3.model.CompletedUpload - The
 result of the completed upload.
 */
public CompletedUpload uploadStream(S3TransferManager transferManager, String
bucketName, String key) {

    BlockingInputStreamAsyncRequestBody body =
        AsyncRequestBody.forBlockingInputStream(null); // 'null'
 indicates a stream will be provided later.

    Upload upload = transferManager.upload(builder -> builder
        .requestBody(body)
        .putObjectRequest(req -> req.bucket(bucketName).key(key))
        .build());

    // AsyncExampleUtils.randomString() returns a random string up to 100
 characters.
    String randomString = AsyncExampleUtils.randomString();
    logger.info("random string to upload: {}: length={}", randomString,
 randomString.length());

    // Provide the stream of data to be uploaded.
    body.writeInputStream(new ByteArrayInputStream(randomString.getBytes()));

    return upload.completionFuture().join();
}
}
```

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie Prüfsummen, um mit einem Amazon S3 S3-Objekt zu arbeiten, indem Sie AWS SDK

Das folgende Codebeispiel zeigt, wie Sie Prüfsummen verwenden, um mit einem Amazon-S3-Objekt zu arbeiten.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

In den Codebeispielen wird eine Teilmenge der folgenden Importe verwendet.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.net.URISyntaxException;
import java.net.URL;
```

```
import java.nio.ByteBuffer;
import java.nio.file.Paths;
import java.security.DigestInputStream;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Base64;
import java.util.List;
import java.util.Objects;
import java.util.UUID;
```

Geben Sie einen Prüfsummenalgorithmus für die `putObject`-Methode an, wenn Sie [PutObjectRequest erstellen](#).

```
public void putObjectWithChecksum() {
    s3Client.putObject(b -> b
        .bucket(bucketName)
        .key(key)
        .checksumAlgorithm(ChecksumAlgorithm.CRC32),
        RequestBody.fromString("This is a test"));
}
```

Überprüfen Sie die Prüfsumme für die `getObject` Methode, wenn Sie [die GetObjectRequest erstellen](#).

```
public GetObjectResponse getObjectWithChecksum() {
    return s3Client.getObject(b -> b
        .bucket(bucketName)
        .key(key)
        .checksumMode(ChecksumMode.ENABLED))
        .response();
}
```

Berechnen Sie im Voraus eine Prüfsumme für die `putObject`-Methode, wenn Sie [PutObjectRequest erstellen](#).

```
public void putObjectWithPrecalculatedChecksum(String filePath) {
    String checksum = calculateChecksum(filePath, "SHA-256");
```

```
s3Client.putObject((b -> b
    .bucket(bucketName)
    .key(key)
    .checksumSHA256(checksum)),
    RequestBody.fromFile(Paths.get(filePath)));
}
```

Verwenden Sie den [S3 Transfer Manager](#) zusätzlich zu [AWS CRT-basierter S3-Client](#) zur transparenten Durchführung eines mehrteiligen Uploads, wenn die Größe des Inhalts einen Schwellenwert überschreitet. Der Standardschwellenwert beträgt 8 MB.

Sie können einen Prüfsummenalgorithmus angeben, der SDK verwendet werden soll. Standardmäßig SDK verwendet der den CRC32 Algorithmus.

```
public void multipartUploadWithChecksumTm(String filePath) {
    S3TransferManager transferManager = S3TransferManager.create();
    UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
        .putObjectRequest(b -> b
            .bucket(bucketName)
            .key(key)
            .checksumAlgorithm(ChecksumAlgorithm.SHA1))
        .source(Paths.get(filePath))
        .build();
    FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
    fileUpload.completionFuture().join();
    transferManager.close();
}
```

Verwenden Sie den [S3Client API](#) oder (S3 AsyncClient API), um einen mehrteiligen Upload durchzuführen. Wenn Sie eine zusätzliche Prüfsumme angeben, müssen Sie den Algorithmus angeben, der bei der Initiierung des Uploads verwendet werden soll. Sie müssen auch den Algorithmus für jede Teilanforderung angeben und die für jedes Teil nach dem Hochladen berechnete Prüfsumme bereitstellen.

```
public void multipartUploadWithChecksumS3Client(String filePath) {
    ChecksumAlgorithm algorithm = ChecksumAlgorithm.CRC32;

    // Initiate the multipart upload.
    CreateMultipartUploadResponse createMultipartUploadResponse =
    s3Client.createMultipartUpload(b -> b
```



```
        .bucket(bucketName)
        .key(key)
        .checksumAlgorithm(algorithm)); // Checksum specified on
initiation.
    String uploadId = createMultipartUploadResponse.uploadId();

    // Upload the parts of the file.
    int partNumber = 1;
    List<CompletedPart> completedParts = new ArrayList<>();
    ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

    try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
        long fileSize = file.length();
        long position = 0;
        while (position < fileSize) {
            file.seek(position);
            long read = file.getChannel().read(bb);

            bb.flip(); // Swap position and limit before reading from the
buffer.

            UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                .bucket(bucketName)
                .key(key)
                .uploadId(uploadId)
                .checksumAlgorithm(algorithm) // Checksum specified on
each part.

                .partNumber(partNumber)
                .build();

            UploadPartResponse partResponse = s3Client.uploadPart(
                uploadPartRequest,
                RequestBody.fromByteBuffer(bb));

            CompletedPart part = CompletedPart.builder()
                .partNumber(partNumber)
                .checksumCRC32(partResponse.checksumCRC32()) // Provide
the calculated checksum.

                .eTag(partResponse.eTag())
                .build();
            completedParts.add(part);

            bb.clear();
            position += read;
            partNumber++;
        }
    }
```

```
    }  
  } catch (IOException e) {  
    System.err.println(e.getMessage());  
  }  
  
  // Complete the multipart upload.  
  s3Client.completeMultipartUpload(b -> b  
    .bucket(bucketName)  
    .key(key)  
    .uploadId(uploadId)  
  
    .multipartUpload(CompletedMultipartUpload.builder().parts(completedParts).build()));  
}
```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for Java 2.x API-Referenz](#).
 - [CompleteMultipartUpload](#)
 - [CreateMultipartUpload](#)
 - [UploadPart](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Arbeiten Sie mit den Objektintegritätsfunktionen von Amazon S3 mithilfe eines AWS SDK

Das folgende Codebeispiel zeigt, wie Sie mit S3-Objektintegritätsfunktionen arbeiten.

C++

SDK für C++

Note

Es gibt noch mehr dazu auf GitHub. Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Führen Sie ein interaktives Szenario aus, in dem die Objektintegritätsfunktionen von Amazon S3 demonstriert werden.

```
#!/ Routine which runs the S3 object integrity workflow.
/*!
  \param clientConfig: Aws client configuration.
  \return bool: Function succeeded.
*/
bool AwsDoc::S3::s3ObjectIntegrityWorkflow(
    const Aws::S3::S3ClientConfiguration &clientConfiguration) {

    /*
     * Create a large file to be used for multipart uploads.
     */
    if (!createLargeFileIfNotExists()) {
        std::cerr << "Workflow exiting because large file creation failed." <<
std::endl;
        return false;
    }

    Aws::String bucketName = TEST_BUCKET_PREFIX;
    bucketName += Aws::Utils::UUID::RandomUUID();
    bucketName = Aws::Utils::StringUtils::ToLower(bucketName.c_str());

    bucketName.resize(std::min(bucketName.size(), MAX_BUCKET_NAME_LENGTH));

    introductoryExplanations(bucketName);

    if (!AwsDoc::S3::createBucket(bucketName, clientConfiguration)) {
        std::cerr << "Workflow exiting because bucket creation failed." <<
std::endl;
        return false;
    }

    Aws::S3::S3ClientConfiguration s3ClientConfiguration(clientConfiguration);
    std::shared_ptr<Aws::S3::S3Client> client =
Aws::MakeShared<Aws::S3::S3Client>("S3Client", s3ClientConfiguration);

    printAsterisksLine();
    std::cout << "Choose from one of the following checksum algorithms."
        << std::endl;

    for (HASH_METHOD hashMethod = DEFAULT; hashMethod <= SHA256; ++hashMethod) {
```

```
        std::cout << " " << hashMethod << " - " <<
stringForHashMethod(hashMethod)
        << std::endl;
    }

    HASH_METHOD chosenHashMethod = askQuestionForIntRange("Enter an index: ",
DEFAULT,
                                                    SHA256);

    gUseCalculatedChecksum = !askYesNoQuestion(
        "Let the SDK calculate the checksum for you? (y/n) ");

    printAsterisksLine();

    std::cout << "The workflow will now upload a file using PutObject."
        << std::endl;
    std::cout << "Object integrity will be verified using the "
        << stringForHashMethod(chosenHashMethod) << " algorithm."
        << std::endl;
    if (gUseCalculatedChecksum) {
        std::cout
            << "A checksum computed by this workflow will be used for object
integrity verification,"
            << std::endl;
        std::cout << "except for the TransferManager upload." << std::endl;
    } else {
        std::cout
            << "A checksum computed by the SDK will be used for object
integrity verification."
            << std::endl;
    }

    pressEnterToContinue();
    printAsterisksLine();

    std::shared_ptr<Aws::IOStream> inputData =
        Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
            TEST_FILE,
            std::ios_base::in |
            std::ios_base::binary);

    if (!*inputData) {
        std::cerr << "Error unable to read file " << TEST_FILE << std::endl;
    }
}
```

```
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    Hasher hasher;
    HASH_METHOD putObjectHashMethod = chosenHashMethod;
    if (putObjectHashMethod == DEFAULT) {
        putObjectHashMethod = MD5; // MD5 is the default hash method for
PutObject.

        std::cout << "The default checksum algorithm for PutObject is "
            << stringForHashMethod(putObjectHashMethod)
            << std::endl;
    }

    // Demonstrate in code how the hash is computed.
    if (!hasher.calculateObjectHash(*inputData, putObjectHashMethod)) {
        std::cerr << "Error calculating hash for file " << TEST_FILE <<
std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }
    Aws::String key = stringForHashMethod(putObjectHashMethod);
    key += "_";
    key += TEST_FILE_KEY;
    Aws::String localHash = hasher.getBase64HashString();

    // Upload the object with PutObject
    if (!putObjectWithHash(bucketName, key, localHash, putObjectHashMethod,
        inputData, chosenHashMethod == DEFAULT,
        *client)) {
        std::cerr << "Error putting file " << TEST_FILE << " to bucket "
            << bucketName << " with key " << key << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    Aws::String retrievedHash;
    if (!retrieveObjectHash(bucketName, key,
        putObjectHashMethod, retrievedHash,
        nullptr, *client)) {
        std::cerr << "Error getting file " << TEST_FILE << " from bucket "
            << bucketName << " with key " << key << std::endl;
        cleanUp(bucketName, clientConfiguration);
    }
}
```

```
        return false;
    }

    explainPutObjectResults();
    verifyHashingResults(retrievedHash, hasher,
                        "PutObject upload", putObjectHashMethod);

    printAsterisksLine();
    pressEnterToContinue();

    key = "tr_";
    key += stringForHashMethod(chosenHashMethod) + "_" + MULTI_PART_TEST_FILE;

    introductoryTransferManagerUploadExplanations(key);

    HASH_METHOD transferManagerHashMethod = chosenHashMethod;
    if (transferManagerHashMethod == DEFAULT) {
        transferManagerHashMethod = CRC32; // The default hash method for the
TransferManager is CRC32.

        std::cout << "The default checksum algorithm for TransferManager is "
                  << stringForHashMethod(transferManagerHashMethod)
                  << std::endl;
    }

    // Upload the large file using the transfer manager.
    if (!doTransferManagerUpload(bucketName, key, transferManagerHashMethod,
chosenHashMethod == DEFAULT,
                                client)) {
        std::cerr << "Exiting because of an error in doTransferManagerUpload." <<
std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    std::vector<Aws::String> retrievedTransferManagerPartHashes;
    Aws::String retrievedTransferManagerFinalHash;

    // Retrieve all the hashes for the TransferManager upload.
    if (!retrieveObjectHash(bucketName, key,
                            transferManagerHashMethod,
                            retrievedTransferManagerFinalHash,
                            &retrievedTransferManagerPartHashes, *client)) {
```

```
        std::cerr << "Exiting because of an error in retrieveObjectHash for
TransferManager." << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    AwsDoc::S3::Hasher locallyCalculatedFinalHash;
    std::vector<Aws::String> locallyCalculatedPartHashes;

    // Calculate the hashes locally to demonstrate how TransferManager hashes are
    computed.
    if (!calculatePartHashesForFile(transferManagerHashMethod,
MULTI_PART_TEST_FILE,
                                     UPLOAD_BUFFER_SIZE,
                                     locallyCalculatedFinalHash,
                                     locallyCalculatedPartHashes)) {
        std::cerr << "Exiting because of an error in calculatePartHashesForFile."
<< std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    verifyHashingResults(retrievedTransferManagerFinalHash,
        locallyCalculatedFinalHash, "TransferManager upload",
        transferManagerHashMethod,
        retrievedTransferManagerPartHashes,
        locallyCalculatedPartHashes);

    printAsterisksLine();

    key = "mp_";
    key += stringForHashMethod(chosenHashMethod) + "_" + MULTI_PART_TEST_FILE;

    multiPartUploadExplanations(key, chosenHashMethod);

    pressEnterToContinue();

    std::shared_ptr<Aws::IOStream> largeFileInputData =
        Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
                                     MULTI_PART_TEST_FILE,
                                     std::ios_base::in |
                                     std::ios_base::binary);

    if (!largeFileInputData->good()) {
```

```
        std::cerr << "Error unable to read file " << TEST_FILE << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    HASH_METHOD multipartUploadHashMethod = chosenHashMethod;
    if (multipartUploadHashMethod == DEFAULT) {
        multipartUploadHashMethod = MD5; // The default hash method for
multipart uploads is MD5.

        std::cout << "The default checksum algorithm for multipart upload is "
            << stringForHashMethod(putObjectHashMethod)
            << std::endl;
    }

    AwsDoc::S3::Hasher hashData;
    std::vector<Aws::String> partHashes;

    if (!doMultipartUpload(bucketName, key,
        multipartUploadHashMethod,
        largeFileInputData, chosenHashMethod == DEFAULT,
        hashData,
        partHashes,
        *client)) {
        std::cerr << "Exiting because of an error in doMultipartUpload." <<
std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    std::cout << "Finished multipart upload of with hash method " <<
        stringForHashMethod(multipartUploadHashMethod) << std::endl;

    std::cout << "Now we will retrieve the checksums from the server." <<
std::endl;

    retrievedHash.clear();
    std::vector<Aws::String> retrievedPartHashes;
    if (!retrieveObjectHash(bucketName, key,
        multipartUploadHashMethod,
        retrievedHash, &retrievedPartHashes, *client)) {
        std::cerr << "Exiting because of an error in retrieveObjectHash for
multipart." << std::endl;
        cleanUp(bucketName, clientConfiguration);
    }
}
```



```

        return false;
    }

    verifyHashingResults(retrievedHash, hashData, "MultiPart upload",
                        multipartUploadHashMethod,
                        retrievedPartHashes, partHashes);

    printAsterisksLine();

    if (askYesNoQuestion("Would you like to delete the resources created in this
workflow? (y/n)")) {
        return cleanUp(bucketName, clientConfiguration);
    } else {
        std::cout << "The bucket " << bucketName << " was not deleted." <<
std::endl;
        return true;
    }
}

//! Routine which uploads an object to an S3 bucket with different object
integrity hashing methods.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param hashData: The hash value that will be associated with the uploaded
object.
    \param hashMethod: The hashing algorithm to use when calculating the hash
value.
    \param body: The data content of the object being uploaded.
    \param useDefaultHashMethod: A flag indicating whether to use the default hash
method or the one specified in the hashMethod parameter.
    \param client: The S3 client instance used to perform the upload operation.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::putObjectWithHash(const Aws::String &bucket, const Aws::String
&key,
                                const Aws::String &hashData,
                                AwsDoc::S3::HASH_METHOD hashMethod,
                                const std::shared_ptr<Aws::IOStream> &body,
                                bool useDefaultHashMethod,
                                const Aws::S3::S3Client &client) {
    Aws::S3::Model::PutObjectRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);

```

```
    if (!useDefaultHashMethod) {
        if (hashMethod != MD5) {

request.SetChecksumAlgorithm(getChecksumAlgorithmForHashMethod(hashMethod));
        }
    }

    if (gUseCalculatedChecksum) {
        switch (hashMethod) {
            case AwsDoc::S3::MD5:
                request.SetContentMD5(hashData);
                break;
            case AwsDoc::S3::SHA1:
                request.SetChecksumSHA1(hashData);
                break;
            case AwsDoc::S3::SHA256:
                request.SetChecksumSHA256(hashData);
                break;
            case AwsDoc::S3::CRC32:
                request.SetChecksumCRC32(hashData);
                break;
            case AwsDoc::S3::CRC32C:
                request.SetChecksumCRC32C(hashData);
                break;
            default:
                std::cerr << "Unknown hash method." << std::endl;
                return false;
        }
    }
    request.SetBody(body);
    Aws::S3::Model::PutObjectOutcome outcome = client.PutObject(request);
    body->seekg(0, body->beg);
    if (outcome.IsSuccess()) {
        std::cout << "Object successfully uploaded." << std::endl;
    } else {
        std::cerr << "Error uploading object." <<
            outcome.GetError().GetMessage() << std::endl;
    }
    return outcome.IsSuccess();
}

// ! Routine which retrieves the hash value of an object stored in an S3 bucket.
/*!
```

```

\param bucket: The name of the S3 bucket where the object is stored.
\param key: The unique identifier (key) of the object within the S3 bucket.
\param hashMethod: The hashing algorithm used to calculate the hash value of
the object.
\param[out] hashData: The retrieved hash.
\param[out] partHashes: The part hashes if available.
\param client: The S3 client instance used to retrieve the object.
\return bool: Function succeeded.
*/
bool AwsDoc::S3::retrieveObjectHash(const Aws::String &bucket, const Aws::String
&key,
                                     AwsDoc::S3::HASH_METHOD hashMethod,
                                     Aws::String &hashData,
                                     std::vector<Aws::String> *partHashes,
                                     const Aws::S3::S3Client &client) {
    Aws::S3::Model::GetObjectAttributesRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);

    if (hashMethod == MD5) {
        Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
        attributes.push_back(Aws::S3::Model::ObjectAttributes::ETag);
        request.SetObjectAttributes(attributes);

        Aws::S3::Model::GetObjectAttributesOutcome outcome =
client.GetObjectAttributes(
    request);
        if (outcome.IsSuccess()) {
            const Aws::S3::Model::GetObjectAttributesResult &result =
outcome.GetResult();
            hashData = result.GetETag();
        } else {
            std::cerr << "Error retrieving object etag attributes." <<
                outcome.GetError().GetMessage() << std::endl;
            return false;
        }
    } else { // hashMethod != MD5
        Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
        attributes.push_back(Aws::S3::Model::ObjectAttributes::Checksum);
        request.SetObjectAttributes(attributes);

        Aws::S3::Model::GetObjectAttributesOutcome outcome =
client.GetObjectAttributes(
    request);
    }
}

```

```
        if (outcome.IsSuccess()) {
            const Aws::S3::Model::GetObjectAttributesResult &result =
outcome.GetResult();
            switch (hashMethod) {
                case AwsDoc::S3::DEFAULT: // NOLINT(*-branch-clone)
                    break; // Default is not supported.
#pragma clang diagnostic push
#pragma ide diagnostic ignored "UnreachableCode"
                case AwsDoc::S3::MD5:
                    break; // MD5 is not supported.
#pragma clang diagnostic pop
                case AwsDoc::S3::SHA1:
                    hashData = result.GetChecksum().GetChecksumSHA1();
                    break;
                case AwsDoc::S3::SHA256:
                    hashData = result.GetChecksum().GetChecksumSHA256();
                    break;
                case AwsDoc::S3::CRC32:
                    hashData = result.GetChecksum().GetChecksumCRC32();
                    break;
                case AwsDoc::S3::CRC32C:
                    hashData = result.GetChecksum().GetChecksumCRC32C();
                    break;
                default:
                    std::cerr << "Unknown hash method." << std::endl;
                    return false;
            }
        } else {
            std::cerr << "Error retrieving object checksum attributes." <<
                outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (nullptr != partHashes) {
            attributes.clear();
            attributes.push_back(Aws::S3::Model::ObjectAttributes::ObjectParts);
            request.SetObjectAttributes(attributes);
            outcome = client.GetObjectAttributes(request);
            if (outcome.IsSuccess()) {
                const Aws::S3::Model::GetObjectAttributesResult &result =
outcome.GetResult();
                const Aws::Vector<Aws::S3::Model::ObjectPart> parts =
result.GetObjectParts().GetParts();
                for (const Aws::S3::Model::ObjectPart &part: parts) {
```

```
        switch (hashMethod) {
            case AwsDoc::S3::DEFAULT: // Default is not supported.
NOLINT(*-branch-clone)
                break;
            case AwsDoc::S3::MD5: // MD5 is not supported.
                break;
            case AwsDoc::S3::SHA1:
                partHashes->push_back(part.GetChecksumSHA1());
                break;
            case AwsDoc::S3::SHA256:
                partHashes->push_back(part.GetChecksumSHA256());
                break;
            case AwsDoc::S3::CRC32:
                partHashes->push_back(part.GetChecksumCRC32());
                break;
            case AwsDoc::S3::CRC32C:
                partHashes->push_back(part.GetChecksumCRC32C());
                break;
            default:
                std::cerr << "Unknown hash method." << std::endl;
                return false;
        }
    }
} else {
    std::cerr << "Error retrieving object attributes for object
parts." <<
        outcome.GetError().GetMessage() << std::endl;
    return false;
}
}
}

return true;
}

//! Verifies the hashing results between the retrieved and local hashes.
/*!
 \param retrievedHash The hash value retrieved from the remote source.
 \param localHash The hash value calculated locally.
 \param uploadtype The type of upload (e.g., "multipart", "single-part").
 \param hashMethod The hashing method used (e.g., MD5, SHA-256).
 \param retrievedPartHashes (Optional) The list of hashes for the individual
parts retrieved from the remote source.
```

```

\param localPartHashes (Optional) The list of hashes for the individual parts
calculated locally.
*/
void AwsDoc::S3::verifyHashingResults(const Aws::String &retrievedHash,
                                     const Hasher &localHash,
                                     const Aws::String &uploadtype,
                                     HASH_METHOD hashMethod,
                                     const std::vector<Aws::String>
&retrievedPartHashes,
                                     const std::vector<Aws::String>
&localPartHashes) {
    std::cout << "For " << uploadtype << " retrieved hash is " << retrievedHash
<< std::endl;
    if (!retrievedPartHashes.empty()) {
        std::cout << retrievedPartHashes.size() << " part hash(es) were also
retrieved."
                << std::endl;
        for (auto &retrievedPartHash: retrievedPartHashes) {
            std::cout << " Part hash " << retrievedPartHash << std::endl;
        }
    }
    Aws::String hashString;
    if (hashMethod == MD5) {
        hashString = localHash.getHexHashString();
        if (!localPartHashes.empty()) {
            hashString += "-" + std::to_string(localPartHashes.size());
        }
    } else {
        hashString = localHash.getBase64HashString();
    }

    bool allMatch = true;
    if (hashString != retrievedHash) {
        std::cerr << "For " << uploadtype << ", the main hashes do not match" <<
std::endl;
        std::cerr << "Local hash- " << hashString << "" << std::endl;
        std::cerr << "Remote hash - " << retrievedHash << "" << std::endl;
        allMatch = false;
    }

    if (hashMethod != MD5) {
        if (localPartHashes.size() != retrievedPartHashes.size()) {
            std::cerr << "For " << uploadtype << ", the number of part hashes do
not match" << std::endl;

```

```

        std::cerr << "Local number of hashes- '" << localPartHashes.size() <<
        ""
        << std::endl;
        std::cerr << "Remote number of hashes - '"
        << retrievedPartHashes.size()
        << "'" << std::endl;
    }

    for (int i = 0; i < localPartHashes.size(); ++i) {
        if (localPartHashes[i] != retrievedPartHashes[i]) {
            std::cerr << "For " << uploadtype << ", the part hashes do not
match for part " << i + 1
            << "." << std::endl;
            std::cerr << "Local hash- '" << localPartHashes[i] << "'"
            << std::endl;
            std::cerr << "Remote hash - '" << retrievedPartHashes[i] << "'"
            << std::endl;
            allMatch = false;
        }
    }
}

if (allMatch) {
    std::cout << "For " << uploadtype << ", locally and remotely calculated
hashes all match!" << std::endl;
}
}

static void transferManagerErrorCallback(const Aws::Transfer::TransferManager *,
                                        const std::shared_ptr<const
                                        Aws::Transfer::TransferHandle> &,
                                        const
                                        Aws::Client::AWSError<Aws::S3::S3Errors> &err) {
    std::cerr << "Error during transfer: '" << err.GetMessage() << "'" <<
    std::endl;
}

static void transferManagerStatusCallback(const Aws::Transfer::TransferManager *,
                                        const std::shared_ptr<const
                                        Aws::Transfer::TransferHandle> &handle) {
    if (handle->GetStatus() == Aws::Transfer::TransferStatus::IN_PROGRESS) {
        std::cout << "Bytes transferred: " << handle->GetBytesTransferred() <<
        std::endl;
    }
}

```

```
    }
}

//! Routine which uploads an object to an S3 bucket using the AWS C++ SDK's
  Transfer Manager.
/*!
  \param bucket: The name of the S3 bucket where the object will be uploaded.
  \param key: The unique identifier (key) for the object within the S3 bucket.
  \param hashMethod: The hashing algorithm to use when calculating the hash
  value.
  \param useDefaultHashMethod: A flag indicating whether to use the default hash
  method or the one specified in the hashMethod parameter.
  \param client: The S3 client instance used to perform the upload operation.
  \return bool: Function succeeded.
*/
bool
AwsDoc::S3::doTransferManagerUpload(const Aws::String &bucket, const Aws::String
&key,
                                   AwsDoc::S3::HASH_METHOD hashMethod,
                                   bool useDefaultHashMethod,
                                   const std::shared_ptr<Aws::S3::S3Client>
&client) {
    std::shared_ptr<Aws::Utils::Threading::PooledThreadExecutor> executor =
  Aws::MakeShared<Aws::Utils::Threading::PooledThreadExecutor>(
    "executor", 25);
    Aws::Transfer::TransferManagerConfiguration transfer_config(executor.get());
    transfer_config.s3Client = client;
    transfer_config.bufferSize = UPLOAD_BUFFER_SIZE;
    if (!useDefaultHashMethod) {
        if (hashMethod == MD5) {
            transfer_config.computeContentMD5 = true;
        } else {
            transfer_config.checksumAlgorithm =
  getChecksumAlgorithmForHashMethod(
            hashMethod);
        }
    }
    transfer_config.errorCallback = transferManagerErrorCallback;
    transfer_config.transferStatusUpdatedCallback =
  transferManagerStatusCallback;

    std::shared_ptr<Aws::Transfer::TransferManager> transfer_manager =
  Aws::Transfer::TransferManager::Create(
    transfer_config);
}
```



```
std::cout << "Uploading the file..." << std::endl;
std::shared_ptr<Aws::Transfer::TransferHandle> uploadHandle =
transfer_manager->UploadFile(MULTI_PART_TEST_FILE,

    bucket, key,

    "text/plain",

    Aws::Map<Aws::String, Aws::String>());
uploadHandle->WaitUntilFinished();
bool success =
    uploadHandle->GetStatus() ==
Aws::Transfer::TransferStatus::COMPLETED;
if (!success) {
    Aws::Client::AWSError<Aws::S3::S3Errors> err = uploadHandle-
>GetLastError();
    std::cerr << "File upload failed: " << err.GetMessage() << std::endl;
}

return success;
}

//! Routine which calculates the hash values for each part of a file being
uploaded to an S3 bucket.
/*!
    \param hashMethod: The hashing algorithm to use when calculating the hash
values.
    \param fileName: The path to the file for which the part hashes will be
calculated.
    \param bufferSize: The size of the buffer to use when reading the file.
    \param[out] hashDataResult: The Hasher object that will store the concatenated
hash value.
    \param[out] partHashes: The vector that will store the calculated hash values
for each part of the file.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::calculatePartHashesForFile(AwsDoc::S3::HASH_METHOD hashMethod,
                                             const Aws::String &fileName,
                                             size_t bufferSize,
                                             AwsDoc::S3::Hasher &hashDataResult,
                                             std::vector<Aws::String> &partHashes)
{
    std::ifstream fileStream(fileName.c_str(), std::ifstream::binary);
```

```
    fileStream.seekg(0, std::ifstream::end);
    size_t objectSize = fileStream.tellg();
    fileStream.seekg(0, std::ifstream::beg);
    std::vector<unsigned char> totalHashBuffer;
    size_t uploadedBytes = 0;

    while (uploadedBytes < objectSize) {
        std::vector<unsigned char> buffer(bufferSize);
        std::streamsize bytesToRead =
static_cast<std::streamsize>(std::min(buffer.size(), objectSize -
uploadedBytes));
        fileStream.read((char *) buffer.data(), bytesToRead);
        Aws::Utils::Stream::PreallocatedStreamBuf
preallocatedStreamBuf(buffer.data(),

bytesToRead);
        std::shared_ptr<Aws::IOStream> body =
            Aws::MakeShared<Aws::IOStream>("SampleAllocationTag",
                &preallocatedStreamBuf);

        Hasher hasher;
        if (!hasher.calculateObjectHash(*body, hashMethod)) {
            std::cerr << "Error calculating hash." << std::endl;
            return false;
        }
        Aws::String base64HashString = hasher.getBase64HashString();
        partHashes.push_back(base64HashString);

        Aws::Utils::ByteBuffer hashBuffer = hasher.getByteBufferHash();

        totalHashBuffer.insert(totalHashBuffer.end(),
hashBuffer.GetUnderlyingData(),
            hashBuffer.GetUnderlyingData() +
hashBuffer.GetLength());

        uploadedBytes += bytesToRead;
    }

    return hashDataResult.calculateObjectHash(totalHashBuffer, hashMethod);
}

//! Create a multipart upload.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
```

```

    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param client: The S3 client instance used to perform the upload operation.
    \return Aws::String: Upload ID or empty string if failed.
*/
Aws::String
AwsDoc::S3::createMultipartUpload(const Aws::String &bucket, const Aws::String
&key,
                                Aws::S3::Model::ChecksumAlgorithm
checksumAlgorithm,
                                const Aws::S3::S3Client &client) {
    Aws::S3::Model::CreateMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);

    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }

    Aws::S3::Model::CreateMultipartUploadOutcome outcome =
        client.CreateMultipartUpload(request);

    Aws::String uploadID;
    if (outcome.IsSuccess()) {
        uploadID = outcome.GetResult().GetUploadId();
    } else {
        std::cerr << "Error creating multipart upload: " <<
outcome.GetError().GetMessage() << std::endl;
    }

    return uploadID;
}

//! Upload a part to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param partNumber:
    \param checksumAlgorithm: Checksum algorithm, ignored when NOT_SET.
    \param calculatedHash: A data integrity hash to set, depending on the
checksum algorithm,
                                ignored when it is an empty string.
    \param body: An shared_ptr IOStream of the data to be uploaded.
    \param client: The S3 client instance used to perform the upload operation.

```

```

    \return UploadPartOutcome: The outcome.
*/

Aws::S3::Model::UploadPartOutcome AwsDoc::S3::uploadPart(const Aws::String
    &bucket,
                                                    const Aws::String &key,
                                                    const Aws::String
    &uploadID,
                                                    int partNumber,
    Aws::S3::Model::ChecksumAlgorithm checksumAlgorithm,
                                                    const Aws::String
    &calculatedHash,
                                                    const
    std::shared_ptr<Aws::IOStream> &body,
                                                    const Aws::S3::S3Client
    &client) {
    Aws::S3::Model::UploadPartRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);
    request.SetPartNumber(partNumber);
    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }
    request.SetBody(body);

    if (!calculatedHash.empty()) {
        switch (checksumAlgorithm) {
            case Aws::S3::Model::ChecksumAlgorithm::NOT_SET:
                request.SetContentMD5(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32:
                request.SetChecksumCRC32(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32C:
                request.SetChecksumCRC32C(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA1:
                request.SetChecksumSHA1(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA256:
                request.SetChecksumSHA256(calculatedHash);
                break;
        }
    }
}

```



```

    \return CompleteMultipartUploadOutcome: The request outcome.
*/
Aws::S3::Model::CompleteMultipartUploadOutcome
    AwsDoc::S3::completeMultipartUpload(const Aws::String &bucket,

    const Aws::String &key,

    const Aws::String &uploadID,

    const Aws::Vector<Aws::S3::Model::CompletedPart> &parts,

    const Aws::S3::S3Client &client) {
    Aws::S3::Model::CompletedMultipartUpload completedMultipartUpload;
    completedMultipartUpload.SetParts(parts);

    Aws::S3::Model::CompleteMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);
    request.SetMultipartUpload(completedMultipartUpload);

    Aws::S3::Model::CompleteMultipartUploadOutcome outcome =
        client.CompleteMultipartUpload(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error completing multipart upload: " <<
outcome.GetError().GetMessage() << std::endl;
    }
    return outcome;
}

//! Routine which performs a multi-part upload.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param hashMethod: The hashing algorithm to use when calculating the hash
value.
    \param ioStream: An IOStream for the data to be uploaded.
    \param useDefaultHashMethod: A flag indicating whether to use the default
hash method or the one specified in the hashMethod parameter.
    \param[out] hashDataResult: The Hasher object that will store the
concatenated hash value.
    \param[out] partHashes: The vector that will store the calculated hash values
for each part of the file.

```

```

    \param client: The S3 client instance used to perform the upload operation.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::doMultipartUpload(const Aws::String &bucket,
                                   const Aws::String &key,
                                   AwsDoc::S3::HASH_METHOD hashMethod,
                                   const std::shared_ptr<Aws::IOStream>
&ioStream,
                                   bool useDefaultHashMethod,
                                   AwsDoc::S3::Hasher &hashDataResult,
                                   std::vector<Aws::String> &partHashes,
                                   const Aws::S3::S3Client &client) {

    // Get object size.
    ioStream->seekg(0, ioStream->end);
    size_t objectSize = ioStream->tellg();
    ioStream->seekg(0, ioStream->beg);

    Aws::S3::Model::ChecksumAlgorithm checksumAlgorithm =
    Aws::S3::Model::ChecksumAlgorithm::NOT_SET;
    if (!useDefaultHashMethod) {
        if (hashMethod != MD5) {
            checksumAlgorithm = getChecksumAlgorithmForHashMethod(hashMethod);
        }
    }
    Aws::String uploadID = createMultipartUpload(bucket, key, checksumAlgorithm,
    client);
    if (uploadID.empty()) {
        return false;
    }

    std::vector<unsigned char> totalHashBuffer;
    bool uploadSucceeded = true;
    std::streamsize uploadedBytes = 0;
    int partNumber = 1;
    Aws::Vector<Aws::S3::Model::CompletedPart> parts;
    while (uploadedBytes < objectSize) {
        std::cout << "Uploading part " << partNumber << "." << std::endl;

        std::vector<unsigned char> buffer(UPLOAD_BUFFER_SIZE);
        std::streamsize bytesToRead =
    static_cast<std::streamsize>(std::min(buffer.size(),
    objectSize - uploadedBytes));
        ioStream->read((char *) buffer.data(), bytesToRead);

```

```
    Aws::Utils::Stream::PreallocatedStreamBuf
preallocatedStreamBuf(buffer.data(),

bytesToRead);
    std::shared_ptr<Aws::IOStream> body =
        Aws::MakeShared<Aws::IOStream>("SampleAllocationTag",
                                        &preallocatedStreamBuf);

    Hasher hasher;
    if (!hasher.calculateObjectHash(*body, hashMethod)) {
        std::cerr << "Error calculating hash." << std::endl;
        uploadSucceeded = false;
        break;
    }

    Aws::String base64HashString = hasher.getBase64HashString();
    partHashes.push_back(base64HashString);

    Aws::Utils::ByteBuffer hashBuffer = hasher.getBytesBufferHash();

    totalHashBuffer.insert(totalHashBuffer.end(),
hashBuffer.GetUnderlyingData(),
                        hashBuffer.GetUnderlyingData() +
hashBuffer.GetLength());

    Aws::String calculatedHash;
    if (gUseCalculatedChecksum) {
        calculatedHash = base64HashString;
    }
    Aws::S3::Model::UploadPartOutcome uploadPartOutcome = uploadPart(bucket,
key, uploadID, partNumber,

checksumAlgorithm, base64HashString, body,

                                                                    client);

    if (uploadPartOutcome.IsSuccess()) {
        const Aws::S3::Model::UploadPartResult &uploadPartResult =
uploadPartOutcome.GetResult();
        Aws::S3::Model::CompletedPart completedPart;
        completedPart.SetETag(uploadPartResult.GetETag());
        completedPart.SetPartNumber(partNumber);
        switch (hashMethod) {
            case AwsDoc::S3::MD5:
                break; // Do nothing.
            case AwsDoc::S3::SHA1:
```



```
completedPart.SetChecksumSHA1(uploadPartResult.GetChecksumSHA1());
    break;
    case AwsDoc::S3::SHA256:

completedPart.SetChecksumSHA256(uploadPartResult.GetChecksumSHA256());
    break;
    case AwsDoc::S3::CRC32:

completedPart.SetChecksumCRC32(uploadPartResult.GetChecksumCRC32());
    break;
    case AwsDoc::S3::CRC32C:

completedPart.SetChecksumCRC32C(uploadPartResult.GetChecksumCRC32C());
    break;
    default:
        std::cerr << "Unhandled hash method for completedPart." <<
std::endl;
        break;
    }

    parts.push_back(completedPart);
} else {
    std::cerr << "Error uploading part. " <<
        uploadPartOutcome.GetError().GetMessage() << std::endl;
    uploadSucceeded = false;
    break;
}

uploadedBytes += bytesToRead;
partNumber++;
}

if (!uploadSucceeded) {
    abortMultipartUpload(bucket, key, uploadID, client);
    return false;
} else {

    Aws::S3::Model::CompleteMultipartUploadOutcome
completeMultipartUploadOutcome = completeMultipartUpload(bucket,

        key,

        uploadID,
```

```
        parts,
        client);

    if (completeMultipartUploadOutcome.IsSuccess()) {
        std::cout << "Multipart upload completed." << std::endl;
        if (!hashDataResult.calculateObjectHash(totalHashBuffer, hashMethod))
    {
            std::cerr << "Error calculating hash." << std::endl;
            return false;
        }
    } else {
        std::cerr << "Error completing multipart upload." <<
            completeMultipartUploadOutcome.GetError().GetMessage()
            << std::endl;
    }

    return completeMultipartUploadOutcome.IsSuccess();
}

}

//! Routine which retrieves the string for a HASH_METHOD constant.
/*!
    \param: hashMethod: A HASH_METHOD constant.
    \return: String: A string description of the hash method.
*/
Aws::String AwsDoc::S3::stringForHashMethod(AwsDoc::S3::HASH_METHOD hashMethod) {
    switch (hashMethod) {
        case AwsDoc::S3::DEFAULT:
            return "Default";
        case AwsDoc::S3::MD5:
            return "MD5";
        case AwsDoc::S3::SHA1:
            return "SHA1";
        case AwsDoc::S3::SHA256:
            return "SHA256";
        case AwsDoc::S3::CRC32:
            return "CRC32";
        case AwsDoc::S3::CRC32C:
            return "CRC32C";
        default:
            return "Unknown";
    }
}
```

```
}

//! Routine that returns the ChecksumAlgorithm for a HASH_METHOD constant.
/*!
  \param hashMethod: A HASH_METHOD constant.
  \return ChecksumAlgorithm: The ChecksumAlgorithm enum.
*/
Aws::S3::Model::ChecksumAlgorithm
AwsDoc::S3::getChecksumAlgorithmForHashMethod(AwsDoc::S3::HASH_METHOD hashMethod)
{
  Aws::S3::Model::ChecksumAlgorithm result =
  Aws::S3::Model::ChecksumAlgorithm::NOT_SET;
  switch (hashMethod) {
    case AwsDoc::S3::DEFAULT:
      std::cerr << "getChecksumAlgorithmForHashMethod- DEFAULT is not
valid." << std::endl;
      break; // Default is not supported.
    case AwsDoc::S3::MD5:
      break; // Ignore MD5.
    case AwsDoc::S3::SHA1:
      result = Aws::S3::Model::ChecksumAlgorithm::SHA1;
      break;
    case AwsDoc::S3::SHA256:
      result = Aws::S3::Model::ChecksumAlgorithm::SHA256;
      break;
    case AwsDoc::S3::CRC32:
      result = Aws::S3::Model::ChecksumAlgorithm::CRC32;
      break;
    case AwsDoc::S3::CRC32C:
      result = Aws::S3::Model::ChecksumAlgorithm::CRC32C;
      break;
    default:
      std::cerr << "Unknown hash method." << std::endl;
      break;

  }

  return result;
}

//! Routine which cleans up after the example is complete.
/*!
  \param bucket: The name of the S3 bucket where the object was uploaded.
  \param clientConfiguration: The client configuration for the S3 client.
```

```
\return bool: Function succeeded.
*/
bool AwsDoc::S3::cleanUp(const Aws::String &bucketName,
                        const Aws::S3::S3ClientConfiguration
                        &clientConfiguration) {

    Aws::Vector<Aws::String> keysResult;
    bool result = true;
    if (AwsDoc::S3::listObjects(bucketName, keysResult, clientConfiguration)) {
        if (!keysResult.empty()) {
            result = AwsDoc::S3::deleteObjects(keysResult, bucketName,
                                              clientConfiguration);
        }
    } else {
        result = false;
    }

    return result && AwsDoc::S3::deleteBucket(bucketName, clientConfiguration);
}

//! Console interaction introducing the workflow.
/*!
\param bucketName: The name of the S3 bucket to use.
*/
void AwsDoc::S3::introductoryExplanations(const Aws::String &bucketName) {

    std::cout
        << "Welcome to the Amazon Simple Storage Service (Amazon S3) object
integrity workflow."
        << std::endl;
    printAsterisksLine();
    std::cout
        << "This workflow demonstrates how Amazon S3 uses checksum values to
verify the integrity of data\n";
    std::cout << "uploaded to Amazon S3 buckets" << std::endl;
    std::cout
        << "The AWS SDK for C++ automatically handles checksums.\n";
    std::cout
        << "By default it calculates a checksum that is uploaded with an
object.\n"
        << "The default checksum algorithm for PutObject and MultiPart upload
is an MD5 hash.\n"
        << "The default checksum algorithm for TransferManager uploads is a
CRC32 checksum."
```

```
        << std::endl;
    std::cout
        << "You can override the default behavior, requiring one of the
following checksums,\n";
    std::cout << "MD5, CRC32, CRC32C, SHA-1 or SHA-256." << std::endl;
    std::cout << "You can also set the checksum hash value, instead of letting
the SDK calculate the value."
        << std::endl;
    std::cout
        << "For more information, see https://docs.aws.amazon.com/AmazonS3/
latest/userguide/checking-object-integrity.html."
        << std::endl;

    std::cout
        << "This workflow will locally compute checksums for files uploaded
to an Amazon S3 bucket,\n";
    std::cout << "even when the SDK also computes the checksum." << std::endl;
    std::cout
        << "This is done to provide demonstration code for how the checksums
are calculated."
        << std::endl;
    std::cout << "A bucket named '" << bucketName << "' will be created for the
object uploads."
        << std::endl;
}

//! Console interaction which explains the PutObject results.
/*!
*/
void AwsDoc::S3::explainPutObjectResults() {

    std::cout << "The upload was successful.\n";
    std::cout << "If the checksums had not matched, the upload would have
failed."
        << std::endl;
    std::cout
        << "The checksums calculated by the server have been retrieved using
the GetObjectAttributes."
        << std::endl;
    std::cout
        << "The locally calculated checksums have been verified against the
retrieved checksums."
        << std::endl;
}
```

```
//! Console interaction explaining transfer manager uploads.
/*!
 \param objectKey: The key for the object being uploaded.
 */
void AwsDoc::S3::introductoryTransferManagerUploadExplanations(
    const Aws::String &objectKey) {
    std::cout
        << "Now the workflow will demonstrate object integrity for
TransferManager multi-part uploads."
        << std::endl;
    std::cout
        << "The AWS C++ SDK has a TransferManager class which simplifies
multipart uploads."
        << std::endl;
    std::cout
        << "The following code lets the TransferManager handle much of the
checksum configuration."
        << std::endl;

    std::cout << "An object with the key '" << objectKey
        << " will be uploaded by the TransferManager using a "
        << BUFFER_SIZE_IN_MEGABYTES << " MB buffer." << std::endl;
    if (gUseCalculatedChecksum) {
        std::cout << "For TransferManager uploads, this demo always lets the SDK
calculate the hash value."
            << std::endl;
    }

    pressEnterToContinue();
    printAsterisksLine();
}

//! Console interaction explaining multi-part uploads.
/*!
 \param objectKey: The key for the object being uploaded.
 \param chosenHashMethod: The hash method selected by the user.
 */
void AwsDoc::S3::multiPartUploadExplanations(const Aws::String &objectKey,
                                             HASH_METHOD chosenHashMethod) {

    std::cout
        << "Now we will provide an in-depth demonstration of multi-part
uploading by calling the multi-part upload APIs directly."
        << std::endl;
```

```

    std::cout << "These are the same APIs used by the TransferManager when
uploading large files."
        << std::endl;
    std::cout
        << "In the following code, the checksums are also calculated locally
and then compared."
        << std::endl;
    std::cout
        << "For multi-part uploads, a checksum is uploaded with each part.
The final checksum is a concatenation of"
        << std::endl;
    std::cout << "the checksums for each part." << std::endl;
    std::cout
        << "This is explained in the user guide, https://docs.aws.amazon.com/
AmazonS3/latest/userguide/checking-object-integrity.html,"
        << " in the section \"Using part-level checksums for multipart
uploads\"." << std::endl;

    std::cout << "Starting multipart upload of with hash method " <<
        stringForHashMethod(chosenHashMethod) << " uploading to with object
key\n"
        << "\"" << objectKey << "\", " << std::endl;
}

//! Create a large file for doing multi-part uploads.
/*!
*/
bool AwsDoc::S3::createLargeFileIfNotExists() {
    // Generate a large file by writing this source file multiple times to a new
file.
    if (std::filesystem::exists(MULTI_PART_TEST_FILE)) {
        return true;
    }

    std::ofstream newFile(MULTI_PART_TEST_FILE, std::ios::out
                            | std::ios::binary);

    if (!newFile) {
        std::cerr << "createLargeFileIfNotExists- Error creating file " <<
MULTI_PART_TEST_FILE <<
            std::endl;
        return false;
    }
}

```

```
    }

    std::ifstream input(TEST_FILE, std::ios::in
                        | std::ios::binary);

    if (!input) {
        std::cerr << "Error opening file " << TEST_FILE <<
                  std::endl;
        return false;
    }
    std::stringstream buffer;
    buffer << input.rdbuf();

    input.close();

    while (newFile.tellp() < LARGE_FILE_SIZE && !newFile.bad()) {
        buffer.seekg(std::stringstream::beg);
        newFile << buffer.rdbuf();
    }

    newFile.close();

    return true;
}
```

- API-Einheiten finden Sie in den folgenden Themen unter [AWS SDK for C++ API-Referenz](#).
 - [AbortMultipartUpload](#)
 - [CompleteMultipartUpload](#)
 - [CreateMultipartUpload](#)
 - [DeleteObject](#)
 - [GetObjectAttributes](#)
 - [PutObject](#)
 - [UploadPart](#)

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Arbeiten Sie mit Objekten mit Amazon S3 S3-Version unter Verwendung eines AWS SDK

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen Sie einen versionierten S3 Bucket.
- Rufen Sie alle Versionen eines Objekts ab.
- Setzen Sie eine Richtlinie auf eine frühere Version zurück.
- Löschen Sie ein versioniertes Objekt und stellen Sie es wieder her.
- Löschen Sie alle Versionen eines Objekts dauerhaft.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Finden Sie das vollständige Beispiel und erfahren Sie, wie Sie es einrichten und ausführen in der [AWS Repository mit Codebeispielen](#).

Erstellen Sie Funktionen, die S3-Aktionen umschließen.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
    lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
```

```
:return: The newly created bucket.
"""
try:
    bucket = s3.create_bucket(
        Bucket=bucket_name,
        CreateBucketConfiguration={
            "LocationConstraint": s3.meta.client.meta.region_name
        },
    )
    logger.info("Created bucket %s.", bucket.name)
except ClientError as error:
    if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
        logger.warning("Bucket %s already exists! Using it.", bucket_name)
        bucket = s3.Bucket(bucket_name)
    else:
        logger.exception("Couldn't create bucket %s.", bucket_name)
        raise

try:
    bucket.Versioning().enable()
    logger.info("Enabled versioning on bucket %s.", bucket.name)
except ClientError:
    logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
    raise

try:
    expiration = 7
    bucket.LifecycleConfiguration().put(
        LifecycleConfiguration={
            "Rules": [
                {
                    "Status": "Enabled",
                    "Prefix": prefix,
                    "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration}],
                }
            ]
        }
    )
    logger.info(
        "Configured lifecycle to expire noncurrent versions after %s days "
        "on bucket %s.",
        expiration,
        bucket.name,
```

```
    )
except ClientError as error:
    logger.warning(
        "Couldn't configure lifecycle on bucket %s because %s. "
        "Continuing anyway.",
        bucket.name,
        error,
    )

return bucket

def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(
        bucket.object_versions.filter(Prefix=object_key),
        key=attrgetter("last_modified"),
        reverse=True,
    )

    logger.debug(
        "Got versions:\n%s",
        "\n".join(
            [
                f"\t{version.version_id}, last modified {version.last_modified}"
                for version in versions
            ]
        ),
    )

    if version_id in [ver.version_id for ver in versions]:
```

```
print(f"Rolling back to version {version_id}")
for version in versions:
    if version.version_id != version_id:
        version.delete()
        print(f"Deleted version {version.version_id}")
    else:
        break

print(f"Active version is now {bucket.Object(object_key).version_id}")
else:
    raise KeyError(
        f"{version_id} was not found in the list of versions for "
        f"{object_key}."
    )

def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
    marker.
    By removing the delete marker, we make the previous version the latest
    version
    and the object then presents as not deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1
    )

    if "DeleteMarkers" in response:
        latest_version = response["DeleteMarkers"][0]
        if latest_version["IsLatest"]:
            logger.info(
                "Object %s was indeed deleted on %s. Let's revive it.",
                object_key,
```

```
        latest_version["LastModified"],
    )
    obj = bucket.Object(object_key)
    obj.Version(latest_version["VersionId"]).delete()
    logger.info(
        "Revived %s, active version is now %s with body '%s'",
        object_key,
        obj.version_id,
        obj.get()["Body"].read(),
    )
else:
    logger.warning(
        "Delete marker is not the latest version for %s!", object_key
    )
elif "Versions" in response:
    logger.warning("Got an active version for %s, nothing to do.",
object_key)
else:
    logger.error("Couldn't get any version info for %s.", object_key)

def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
    module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Laden Sie die Strophe eines Gedichts in ein versioniertes Objekt hoch und führen Sie eine Reihe von Aktionen dafür aus.

```
def usage_demo_single_object(obj_prefix="demo-versioning/"):
    """
    Demonstrates usage of versioned object functions. This demo uploads a stanza
    of a poem and performs a series of revisions, deletions, and revivals on it.

    :param obj_prefix: The prefix to assign to objects created by this demo.
    """
    with open("father_william.txt") as file:
        stanzas = file.read().split("\n\n")

    width = get_terminal_size((80, 20))[0]
    print("-" * width)
    print("Welcome to the usage demonstration of Amazon S3 versioning.")
    print(
        "This demonstration uploads a single stanza of a poem to an Amazon "
        "S3 bucket and then applies various revisions to it."
    )
    print("-" * width)
    print("Creating a version-enabled bucket for the demo...")
    bucket = create_versioned_bucket("bucket-" + str(uuid.uuid1()), obj_prefix)

    print("\nThe initial version of our stanza:")
    print(stanzas[0])

    # Add the first stanza and revise it a few times.
    print("\nApplying some revisions to the stanza...")
    obj_stanza_1 = bucket.Object(f"{obj_prefix}stanza-1")
    obj_stanza_1.put(Body=bytes(stanzas[0], "utf-8"))
    obj_stanza_1.put(Body=bytes(stanzas[0].upper(), "utf-8"))
    obj_stanza_1.put(Body=bytes(stanzas[0].lower(), "utf-8"))
    obj_stanza_1.put(Body=bytes(stanzas[0][::-1], "utf-8"))
    print(
        "The latest version of the stanza is now:",
        obj_stanza_1.get()["Body"].read().decode("utf-8"),
        sep="\n",
    )

    # Versions are returned in order, most recent first.
    obj_stanza_1_versions =
    bucket.object_versions.filter(Prefix=obj_stanza_1.key)
```

```
print(
    "The version data of the stanza revisions:",
    *[
        f"    {version.version_id}, last modified {version.last_modified}"
        for version in obj_stanza_1_versions
    ],
    sep="\n",
)

# Rollback two versions.
print("\nRolling back two versions...")
rollback_object(bucket, obj_stanza_1.key, list(obj_stanza_1_versions)
[2].version_id)
print(
    "The latest version of the stanza:",
    obj_stanza_1.get()["Body"].read().decode("utf-8"),
    sep="\n",
)

# Delete the stanza
print("\nDeleting the stanza...")
obj_stanza_1.delete()
try:
    obj_stanza_1.get()
except ClientError as error:
    if error.response["Error"]["Code"] == "NoSuchKey":
        print("The stanza is now deleted (as expected).")
    else:
        raise

# Revive the stanza
print("\nRestoring the stanza...")
revive_object(bucket, obj_stanza_1.key)
print(
    "The stanza is restored! The latest version is again:",
    obj_stanza_1.get()["Body"].read().decode("utf-8"),
    sep="\n",
)

# Permanently delete all versions of the object. This cannot be undone!
print("\nPermanently deleting all versions of the stanza...")
permanently_delete_object(bucket, obj_stanza_1.key)
obj_stanza_1_versions =
bucket.object_versions.filter(Prefix=obj_stanza_1.key)
```

```
if len(list(obj_stanza_1_versions)) == 0:
    print("The stanza has been permanently deleted and now has no versions.")
else:
    print("Something went wrong. The stanza still exists!")

print(f"\nRemoving {bucket.name}...")
bucket.delete()
print(f"{bucket.name} deleted.")
print("Demo done!")
```

- API Einzelheiten finden Sie in den folgenden Themen unter AWS SDK für Python (Boto3) API-Referenz.
 - [CreateBucket](#)
 - [DeleteObject](#)
 - [ListObjectVersions](#)
 - [PutBucketLifecycleConfiguration](#)

Für eine vollständige Liste von AWS SDK Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK Versionen.

Serverlose Beispiele für Amazon S3 mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Amazon S3 verwendet wird mit AWS SDKs.

Beispiele

- [Aufrufen einer Lambda-Funktion über einen Amazon-S3-Auslöser](#)

Aufrufen einer Lambda-Funktion über einen Amazon-S3-Auslöser

In den folgenden Codebeispielen wird die Implementierung einer Lambda-Funktion gezeigt, die ein Ereignis empfängt, das durch Hochladen eines Objekts in einen S3-Bucket ausgelöst wird. Die Funktion ruft den S3-Bucket-Namen und den Objektschlüssel aus dem Event-Parameter ab und ruft Amazon S3 API auf, um den Inhaltstyp des Objekts abzurufen und zu protokollieren.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit [Serverless-Beispielen](#).

Konsumieren eines S3-Ereignisses mit Lambda unter Verwendung. NET.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
using System.Threading.Tasks;
using Amazon.Lambda.Core;
using Amazon.S3;
using System;
using Amazon.Lambda.S3Events;
using System.Web;

// Assembly attribute to enable the Lambda function's JSON input to be converted
// into a .NET class.
[assembly:
    LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambdaJsonSerializer))]

namespace S3Integration
{
    public class Function
    {
        private static AmazonS3Client _s3Client;
        public Function() : this(null)
        {
        }

        internal Function(AmazonS3Client s3Client)
        {
            _s3Client = s3Client ?? new AmazonS3Client();
        }

        public async Task<string> Handler(S3Event evt, ILambdaContext context)
        {
            try
```

```
    {
        if (evt.Records.Count <= 0)
        {
            context.Logger.LogLine("Empty S3 Event received");
            return string.Empty;
        }

        var bucket = evt.Records[0].S3.Bucket.Name;
        var key = HttpUtility.UrlDecode(evt.Records[0].S3.Object.Key);

        context.Logger.LogLine($"Request is for {bucket} and {key}");

        var objectResult = await _s3Client.GetObjectAsync(bucket, key);


        context.Logger.LogLine($"Returning {objectResult.Key}");

        return objectResult.Key;
    }
    catch (Exception e)
    {
        context.Logger.LogLine($"Error processing request -
{e.Message}");

        return string.Empty;
    }
}
}
```

Go

SDK für Go V2

 Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit [Serverless-Beispielen](#).

Nutzen eines S3-Ereignisses mit Lambda unter Verwendung von Go

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
// SPDX-License-Identifier: Apache-2.0
package main

import (
    "context"
    "log"

    "github.com/aws/aws-lambda-go/events"
    "github.com/aws/aws-lambda-go/lambda"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
)

func handler(ctx context.Context, s3Event events.S3Event) error {
    sdkConfig, err := config.LoadDefaultConfig(ctx)
    if err != nil {
        log.Printf("failed to load default config: %s", err)
        return err
    }
    s3Client := s3.NewFromConfig(sdkConfig)

    for _, record := range s3Event.Records {
        bucket := record.S3.Bucket.Name
        key := record.S3.Object.URLDecodedKey
        headOutput, err := s3Client.HeadObject(ctx, &s3.HeadObjectInput{
            Bucket: &bucket,
            Key:    &key,
        })
        if err != nil {
            log.Printf("error getting head of object %s/%s: %s", bucket, key, err)
            return err
        }
        log.Printf("successfully retrieved %s/%s of type %s", bucket, key,
            *headOutput.ContentType)
    }

    return nil
}

func main() {
    lambda.Start(handler)
}
```

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu. GitHub Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit [Serverless-Beispielen](#).

Nutzen eines S3-Ereignisses mit Lambda unter Verwendung von Java

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
package example;

import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.S3Client;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.S3Event;
import
    com.amazonaws.services.lambda.runtime.events.models.s3.S3EventNotification.S3EventNotifi

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Handler implements RequestHandler<S3Event, String> {
    private static final Logger logger = LoggerFactory.getLogger(Handler.class);
    @Override
    public String handleRequest(S3Event s3event, Context context) {
        try {
            S3EventNotificationRecord record = s3event.getRecords().get(0);
            String srcBucket = record.getS3().getBucket().getName();
            String srcKey = record.getS3().getObject().getUrlDecodedKey();

            S3Client s3Client = S3Client.builder().build();
            HeadObjectResponse headObject = getHeadObject(s3Client, srcBucket,
srcKey);
```

```
        logger.info("Successfully retrieved " + srcBucket + "/" + srcKey + " of
type " + headObject.contentType());

        return "Ok";
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

private HeadObjectResponse getHeadObject(S3Client s3Client, String bucket,
String key) {
    HeadObjectRequest headObjectRequest = HeadObjectRequest.builder()
        .bucket(bucket)
        .key(key)
        .build();
    return s3Client.headObject(headObjectRequest);
}
}
```

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu [GitHub](#). Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit [Serverless-Beispielen](#).

Konsumieren eines S3-Ereignisses mit Lambda unter Verwendung JavaScript.

```
import { S3Client, HeadObjectCommand } from "@aws-sdk/client-s3";

const client = new S3Client();

export const handler = async (event, context) => {

    // Get the object from the event and show its content type
    const bucket = event.Records[0].s3.bucket.name;
    const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g,
' '));
```

```
    try {
      const { ContentType } = await client.send(new HeadObjectCommand({
        Bucket: bucket,
        Key: key,
      }));

      console.log('CONTENT TYPE:', ContentType);
      return ContentType;

    } catch (err) {
      console.log(err);
      const message = `Error getting object ${key} from bucket ${bucket}. Make
sure they exist and your bucket is in the same region as this function.`;
      console.log(message);
      throw new Error(message);
    }
  };
```

Konsumieren eines S3-Ereignisses mit Lambda unter Verwendung TypeScript.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { S3Event } from 'aws-lambda';
import { S3Client, HeadObjectCommand } from '@aws-sdk/client-s3';

const s3 = new S3Client({ region: process.env.AWS_REGION });

export const handler = async (event: S3Event): Promise<string | undefined> => {
  // Get the object from the event and show its content type
  const bucket = event.Records[0].s3.bucket.name;
  const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, '
'));
  const params = {
    Bucket: bucket,
    Key: key,
  };
  try {
    const { ContentType } = await s3.send(new HeadObjectCommand(params));
    console.log('CONTENT TYPE:', ContentType);
    return ContentType;
  } catch (err) {
    console.log(err);
  }
};
```

```
const message = `Error getting object ${key} from bucket ${bucket}. Make sure they exist and your bucket is in the same region as this function.`;
console.log(message);
throw new Error(message);
}
};
```

PHP

SDK für PHP

Note

Es gibt noch mehr dazu. [GitHub](#) Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit [Serverless-Beispielen](#).

Konsumieren eines S3-Ereignisses mit Lambda unter Verwendung PHP.

```
<?php

use Bref\Context\Context;
use Bref\Event\S3\S3Event;
use Bref\Event\S3\S3Handler;
use Bref\Logger\StderrLogger;

require __DIR__ . '/vendor/autoload.php';

class Handler extends S3Handler
{
    private StderrLogger $logger;
    public function __construct(StderrLogger $logger)
    {
        $this->logger = $logger;
    }

    public function handleS3(S3Event $event, Context $context) : void
    {
        $this->logger->info("Processing S3 records");

        // Get the object from the event and show its content type
```

```
$records = $event->getRecords();

foreach ($records as $record)
{
    $bucket = $record->getBucket()->getName();
    $key = urldecode($record->getObject()->getKey());

    try {
        $fileSize = urldecode($record->getObject()->getSize());
        echo "File Size: " . $fileSize . "\n";
        // TODO: Implement your custom processing logic here
    } catch (Exception $e) {
        echo $e->getMessage() . "\n";
        echo 'Error getting object ' . $key . ' from bucket ' .
        $bucket . '. Make sure they exist and your bucket is in the same region as this
        function.' . "\n";
        throw $e;
    }
}

$logger = new StderrLogger();
return new Handler($logger);
```

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit [Serverless-Beispielen](#).

Nutzen eines S3-Ereignisses mit Lambda unter Verwendung von Python

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import json
import urllib.parse
```



```
import boto3

print('Loading function')

s3 = boto3.client('s3')

def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
    encoding='utf-8')
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and
        your bucket is in the same region as this function.'.format(key, bucket))
        raise e
```

Ruby

SDK für Ruby

Note

Es gibt noch mehr dazu GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit [Serverless-Beispielen](#).

Ein S3-Ereignis mit Lambda mithilfe von Ruby konsumieren.

```
require 'json'
require 'uri'
require 'aws-sdk'

puts 'Loading function'
```



```
#[tokio::main]
async fn main() -> Result<(), Error> {
    tracing_subscriber::fmt()
        .with_max_level(tracing::Level::INFO)
        .with_target(false)
        .without_time()
        .init();

    // Initialize the AWS SDK for Rust
    let config = aws_config::load_from_env().await;
    let s3_client = Client::new(&config);

    let res = run(service_fn(|request: LambdaEvent<S3Event>| {
        function_handler(&s3_client, request)
    })).await;

    res
}

async fn function_handler(
    s3_client: &Client,
    evt: LambdaEvent<S3Event>
) -> Result<(), Error> {
    tracing::info!(records = ?evt.payload.records.len(), "Received request from
SQS");

    if evt.payload.records.len() == 0 {
        tracing::info!("Empty S3 event received");
    }

    let bucket = evt.payload.records[0].s3.bucket.name.as_ref().expect("Bucket
name to exist");
    let key = evt.payload.records[0].s3.object.key.as_ref().expect("Object key to
exist");

    tracing::info!("Request is for {} and object {}", bucket, key);

    let s3_get_object_result = s3_client
        .get_object()
        .bucket(bucket)
        .key(key)
        .send()
        .await;
```

```
match s3_get_object_result {
  Ok(_) => tracing::info!("S3 Get Object success, the s3GetObjectResult
contains a 'body' property of type ByteStream"),
  Err(_) => tracing::info!("Failure with S3 Get Object request")
}

Ok(())
}
```

Für eine vollständige Liste von AWS SDK-Entwicklerhandbüchern und Codebeispielen finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Fehlerbehebung

In diesem Abschnitt wird beschrieben, wie Sie Probleme mit Amazon S3 S3-Funktionen beheben können und wie Sie eine Anfrage erhalten IDs, die Sie benötigen, wenn Sie Kontakt aufnehmen AWS Support.

Themen

- [Problembehandlung bei Zugriff verweigert \(403 Forbidden\) Fehler in Amazon S3](#)
- [Fehlerbehebung bei Batch Operations](#)
- [Behebung von Problemen mit Amazon S3 Lifecycle](#)
- [Fehlerbehebung bei einer Replikation](#)
- [Behebung von Fehlern bei der Server-Zugriffsprotokollierung](#)
- [Fehlerbehebung für die Versionsverwaltung](#)
- [Ich erhalte eine Amazon S3 S3-Anfrage IDs für AWS Support](#)

Problembehandlung bei Zugriff verweigert (403 Forbidden) Fehler in Amazon S3


Fehler „Zugriff verweigert“ (HTTP403 Forbidden) treten auf, wenn AWS lehnt eine Autorisierungsanfrage explizit oder implizit ab.

- Eine ausdrückliche Ablehnung liegt vor, wenn eine Richtlinie eine Deny Aussage für einen bestimmten AWS Aktion.
- Eine implizite Verweigerung tritt auf, wenn es keine entsprechende Deny-Anweisung, aber auch keine anwendbare Allow-Anweisung gibt.

Weil ein AWS Identity and Access Management (IAM) Die Richtlinie lehnt standardmäßig implizit einen IAM Prinzipal ab. Die Richtlinie muss dem Prinzipal ausdrücklich erlauben, eine Aktion auszuführen. Andernfalls verweigert die Richtlinie implizit den Zugriff. Weitere Informationen finden Sie [im Benutzerhandbuch unter Der Unterschied zwischen expliziten und impliziten Verweigerungen](#). IAM Informationen zur Richtlinienbewertungslogik, die bestimmt, ob eine Zugriffsanforderung zulässig oder verweigert wird, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM Benutzerhandbuch.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter [Erforderliche Berechtigungen für Amazon S3 API S3-Operationen](#).


Die folgenden Themen behandeln die häufigsten Ursachen für Fehler mit Zugriffsverweigerung in Amazon S3.

 Note

Bei Fehlern mit Zugriff verweigert (HTTP403 Forbidden) berechnet Amazon S3 dem Bucket-Besitzer keine Gebühren, wenn die Anfrage außerhalb der Person des Bucket-Besitzers initiiert wird AWS Konto oder das des Bucket-Besitzers AWS Organisation.

Themen

- [Beispiele für abgelehnte Nachrichten und deren Behebung](#)
- [Bucket-Richtlinien und IAM Richtlinien](#)
- [Amazon S3 ACL S3-Einstellungen](#)
- [S3-Block-Public-Access-Einstellungen](#)
- [Amazon-S3-Verschlüsselungseinstellungen](#)
- [S3-Einstellungen für die Objektsperre](#)
- [VPC-Endpunktrichtlinien](#)
- [AWS Organizations policies](#)
- [Zugriffspunkteinstellungen](#)

 Note

Wenn Sie versuchen, ein Problem mit Berechtigungen zu beheben, beginnen Sie mit dem [the section called “Beispiele für abgelehnte Nachrichten und deren Behebung”](#) Abschnitt und fahren Sie dann mit dem [???](#) Abschnitt fort. Achte außerdem darauf, die Anweisungen unter zu befolgende [the section called “Tipps zum Überprüfen von Berechtigungen”](#).

Beispiele für abgelehnte Nachrichten und deren Behebung

Important

Ab dem 21. August 2024 werden in den kommenden Wochen erweiterte Fehlermeldungen mit dem Status „Zugriff verweigert“ für Anfragen zum selben Konto verfügbar sein. Diese Meldungen werden in allen verfügbaren AWS-Regionen, einschließlich der AWS GovCloud (US) Regions und die Regionen Chinas.

Die meisten Zugriffsverweigerungs-Fehlermeldungen haben das Format User *user-arn* is not authorized to perform *action* on "*resource-arn*" because *context*. In diesem Beispiel: *Benutzer-ARN* ist der [Amazon-Ressourcenname \(ARN\)](#) des Benutzers, der keinen Zugriff erhält, *action* ist die Serviceaktion, die die Richtlinie verweigert, und *resource-arn* ist die ARN Ressource, auf die sich die Richtlinie bezieht. Das Tool *context* Das Feld steht für zusätzlichen Kontext zum Richtlinientyp, der erklärt, warum die Richtlinie den Zugriff verweigert hat.

Wenn eine Richtlinie den Zugriff ausdrücklich verweigert, weil die Richtlinie eine Deny Erklärung enthält, enthält die Fehlermeldung „Zugriff verweigert“ den folgenden Satz with an explicit deny in a *type* policy. Wenn die Richtlinie implizit den Zugriff verweigert, enthält die Fehlermeldung „Zugriff verweigert“ den folgenden Satz. because no *type* policy allows the *action* action

Important

- Meldungen zur erweiterten Zugriffsverweigerung werden nur bei Anfragen für dasselbe Konto zurückgegeben. Bei kontoübergreifenden Anfragen wird eine generische Nachricht zurückgegeben. Access Denied

Informationen zur Bewertungslogik für Richtlinien, mit der bestimmt wird, ob eine kontenübergreifende Zugriffsanfrage zulässig oder verweigert wird, finden Sie im Benutzerhandbuch unter [Logik der IAM kontenübergreifenden Richtlinienbewertung](#). Eine exemplarische Vorgehensweise, die zeigt, wie Sie kontenübergreifenden Zugriff gewähren, finden Sie unter [the section called "Erteilen von kontenübergreifenden Berechtigungen"](#)

- Für Anfragen an Directory-Buckets werden keine Fehlermeldungen mit dem Status „Erweiterter Zugriff verweigert“ zurückgegeben. Directory-Bucket-Anfragen geben eine generische Access Denied Meldung zurück.

- Wenn mehrere Richtlinien desselben Richtlinientyps eine Autorisierungsanfrage ablehnen, gibt die Fehlermeldung „Zugriff verweigert“ nicht die Anzahl der Richtlinien an.
- Wenn mehrere Richtlinientypen eine Autorisierungsanfrage ablehnen, enthält die Fehlermeldung nur einen dieser Richtlinientypen.
- Wenn eine Zugriffsanfrage aus mehreren Gründen abgelehnt wird, enthält die Fehlermeldung nur einen der Gründe für die Ablehnung.

Die folgenden Beispiele zeigen das Format für die verschiedenen Arten von Fehlermeldungen mit Zugriffsverweigerung und die Problembehandlung der einzelnen Meldungstypen.

Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – implizite Verweigerung

1. Suchen Sie in Ihren Service Control-Richtlinien nach einer fehlenden Allow Aussage für die Aktion (SCPs). Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre, SCP indem Sie die Allow Erklärung hinzufügen. Weitere Informationen finden Sie unter [Aktualisieren eines SCP](#) in der AWS Organizations Benutzerleitfaden.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject because no service control policy allows the s3:GetObject action
```

Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – explizite Verweigerung

1. Suchen Sie in Ihren Service Control-Richtlinien nach einer Deny Erklärung zu der Aktion (SCPs). Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre, SCP indem Sie die Deny Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Ein Beispiel dafür, wie Sie dies tun können, finden Sie unter [IAMVerhindern, dass Benutzer und Rollen bestimmte Änderungen vornehmen, mit einer Ausnahme für eine bestimmte Administratorrolle](#) in der AWS Organizations Benutzerleitfaden. Weitere Informationen zur Aktualisierung Ihres SCP finden Sie unter [Aktualisieren eines SCP](#) in der AWS Organizations Benutzerleitfaden.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject with an explicit deny in a service control policy
```


Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — implizite Ablehnung

1. Suchen Sie in Ihren Virtual Private Cloud (VPC) -Endpunktrichtlinien nach einer fehlenden Allow Angabe für die Aktion. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre VPC Endpunktrichtlinie, indem Sie die Allow Erklärung hinzufügen. Weitere Informationen finden Sie unter [Aktualisieren einer VPC Endpunktrichtlinie](#) im AWS PrivateLink Leitfaden.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no VPC endpoint policy allows the s3:GetObject action
```

Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — ausdrückliche Ablehnung

1. Suchen Sie in Ihren Virtual Private Cloud (VPC) -Endpunktrichtlinien nach einer ausdrücklichen Deny Erklärung für die Aktion. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre VPC Endpunktrichtlinie, indem Sie die Deny Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre Deny Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter [gezeigt](#) [the section called "Beispiel 7: Ausschließen von Hauptpersonen aus Kontoauszügen Deny"](#). Weitere Informationen zur Aktualisierung Ihrer VPC Endpunktrichtlinie finden Sie unter [Aktualisieren einer VPC Endpunktrichtlinie](#) im AWS PrivateLink Leitfaden.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a VPC endpoint policy
```

Zugriff aufgrund einer Berechtigungsgrenze verweigert – implizite Ablehnung

1. Überprüfen Sie, ob in Ihrer Berechtigungsgrenze eine Allow-Anweisung für die Aktion fehlt. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Zugriffsgrenzen, indem Sie die Allow Erklärung zu Ihrer IAM Richtlinie hinzufügen. Weitere Informationen finden Sie unter [Zugriffsgrenzen für IAM Entitäten](#) und [Bearbeiten von IAM Richtlinien](#) im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name"
because no permissions boundary allows the s3:GetObject action
```

Zugriff aufgrund einer Berechtigungsgrenze verweigert – explizite Ablehnung

1. Überprüfen Sie, ob in Ihren Berechtigungsgrenzen eine explizite Deny-Anweisung für die Aktion vorhanden ist. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Berechtigungsgrenze, indem Sie die Deny Aussage in Ihrer IAM Richtlinie ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre Deny Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter [aws:PrincipalAccount](#) im IAM-Benutzerhandbuch. Weitere Informationen finden Sie unter [Zugriffsgrenzen für IAM Entitäten](#) und [IAMBearbeitungsrichtlinien](#) im IAMBenutzerhandbuch.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject with an explicit deny in a permissions boundary
```

Zugriff aufgrund von Sitzungsrichtlinien verweigert – implizite Ablehnung

1. Überprüfen Sie, ob in Ihren Sitzungsrichtlinien eine Allow-Anweisung für die Aktion fehlt. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Sitzungsrichtlinie, indem Sie die Allow-Anweisung hinzufügen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) und [IAMBearbeitungsrichtlinien](#) im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no session policy allows the s3:GetObject action
```

Zugriff aufgrund von Sitzungsrichtlinien verweigert – explizite Ablehnung

1. Überprüfen Sie, ob in Ihren Sitzungsrichtlinien eine explizite Deny-Anweisung für die Aktion vorhanden ist. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.

2. Aktualisieren Sie Ihre Sitzungsrichtlinie, indem Sie die Deny Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre Deny Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter [gezeigt](#) [the section called "Beispiel 7: Ausschließen von Hauptpersonen aus Kontoauszügen Deny"](#). Weitere Informationen zur Aktualisierung Ihrer Sitzungsrichtlinie finden Sie unter [Sitzungsrichtlinien](#) und [IAMBearbeitungsrichtlinien](#) im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a session policy
```

Zugriff aufgrund ressourcenbasierter Richtlinien verweigert – implizite Ablehnung

Note

Unter ressourcenbasierten Richtlinien sind Richtlinien wie Bucket-Richtlinien und Zugriffspunktrichtlinien zu verstehen.

1. Überprüfen Sie, ob in Ihrer ressourcenbasierten Richtlinie eine Allow-Anweisung für die Aktion fehlt. Prüfen Sie auch, ob die Einstellung `IgnorePublicAcls` S3 Block Public Access auf Bucket-, Access Point- oder Kontoebene angewendet wird. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die Allow-Anweisung hinzufügen. Weitere Informationen finden Sie unter [Ressourcenbasierte Richtlinien](#) und [IAMBearbeitungsrichtlinien](#) im IAMBenutzerhandbuch.

Möglicherweise müssen Sie auch Ihre Einstellung „Öffentlichen Zugriff `IgnorePublicAcls` blockieren“ für den Bucket, den Access Point oder das Konto anpassen. Weitere Informationen erhalten Sie unter [the section called "Der Zugriff wurde aufgrund der Einstellungen zum Blockieren des öffentlichen Zugriffs verweigert"](#) und [the section called "Konfigurieren von Bucket- und Zugriffspunktseinstellungen"](#).

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
```

```
s3:GetObject because no resource-based policy allows the s3:GetObject action
```

Zugriff aufgrund ressourcenbasierter Richtlinien verweigert – explizite Ablehnung

Note

Unter ressourcenbasierten Richtlinien sind Richtlinien wie Bucket-Richtlinien und Zugriffspunktrichtlinien zu verstehen.

1. Suchen Sie nach einer expliziten Deny-Anweisung für die Aktion in Ihrer ressourcenbasierten Richtlinie. Prüfen Sie auch, ob die Einstellung `RestrictPublicBuckets S3 Block Public Access` auf Bucket-, Access Point- oder Kontoebene angewendet wird. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die Deny Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre Deny Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter gezeigt [the section called “Beispiel 7: Ausschließen von Hauptpersonen aus Kontoauszügen Deny”](#). Weitere Informationen zur Aktualisierung Ihrer ressourcenbasierten Richtlinie finden Sie unter [Ressourcenbasierte Richtlinien und IAM Bearbeitungsrichtlinien im Benutzerhandbuch](#). IAM

Möglicherweise müssen Sie auch Ihre Einstellung „Öffentlichen Zugriff RestrictPublicBuckets sperren“ für den Bucket, den Access Point oder das Konto anpassen. Weitere Informationen erhalten Sie unter [the section called “Der Zugriff wurde aufgrund der Einstellungen zum Blockieren des öffentlichen Zugriffs verweigert”](#) und [the section called “Konfigurieren von Bucket- und Zugriffspunktseinstellungen”](#).

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform: s3:GetObject on resource: "arn:aws:s3::amzn-s3-demo-bucket1/object-name" with an explicit deny in a resource-based policy
```

Zugriff aufgrund identitätsbasierter Richtlinien verweigert – implizite Verweigerung

1. Überprüfen Sie ob in identitätsbasierten Richtlinien, die der Identität angefügt sind, eine Allow-Anweisung für die Aktion fehlt. Im folgenden Beispiel ist die Aktion an den Benutzer `s3:GetObject` angehängt `MaryMajor`.
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die Allow-Anweisung hinzufügen. Weitere Informationen finden Sie unter [Identitätsbasierte Richtlinien](#) und [IAMBearbeitungsrichtlinien](#) im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no identity-based policy allows the s3:GetObject action
```

Zugriff aufgrund identitätsbasierter Richtlinien verweigert – explizite Ablehnung

1. Überprüfen Sie, ob in identitätsbasierten Richtlinien, die der Identität angefügt sind, eine explizite Deny-Anweisung für die Aktion vorhanden ist. Im folgenden Beispiel ist die Aktion an den Benutzer `s3:GetObject` angehängt. `MaryMajor`
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die Deny Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre Deny Erklärung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter [aws:PrincipalAccount](#) im IAM-Benutzerhandbuch. Weitere Informationen finden Sie unter [Identitätsbasierte Richtlinien](#) und [IAMBearbeitungsrichtlinien](#) im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3::amzn-s3-demo-bucket1/object-name" with
an explicit deny in an identity-based policy
```

Der Zugriff wurde aufgrund der Einstellungen zum Blockieren des öffentlichen Zugriffs verweigert

Die Amazon S3 Block Public Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Weitere Informationen dazu, wie „öffentlich“ in Amazon S3 definiert ist, finden Sie unter [Die Bedeutung von „öffentlich“](#).

Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff. Benutzer können jedoch Bucket-Richtlinien, Zugriffspunktrichtlinien, IAM Benutzerrichtlinien, Objektberechtigungen oder Zugriffskontrolllisten (ACLs) ändern, um den öffentlichen Zugriff zu ermöglichen. Die Einstellungen von S3 Block Public Access haben Vorrang vor diesen Richtlinien, Berechtigungen und ACLs. Seit April 2023 sind alle Einstellungen für Block Public Access standardmäßig für neue Buckets aktiviert.

Wenn Amazon S3 eine Anforderung zum Zugriff auf einen Bucket oder ein Objekt erhält, wird ermittelt, ob für den Bucket oder das Konto des Bucket-Eigentümers eine Block Public Access-Einstellung vorliegt. Wenn die Anforderung über einen Zugriffspunkt einging, prüft Amazon S3 auch auf Block Public Access-Einstellungen für den Zugriffspunkt. Wenn eine Block Public Access-Einstellung vorhanden ist, die den angeforderten Zugriff verbietet, lehnt Amazon S3 die Anforderung ab.

Amazon S3 Block Public Access bietet vier Einstellungen. Diese Einstellungen sind voneinander unabhängig und können in beliebiger Kombination verwendet werden. Jede Einstellung kann auf einen Access Point, einen Bucket oder einen ganzen Bereich angewendet werden AWS Konto. Wenn sich die Block Public Access-Einstellungen für den Zugriffspunkt, den Bucket oder das Konto unterscheiden, wendet Amazon S3 die restriktivste Kombination der Zugriffspunkt-, Bucket- und Kontoeinstellungen an.

Wenn Amazon S3 ermittelt, ob eine Operation von einer Block Public Access-Einstellung untersagt wird, werden alle Anforderungen abgelehnt, die gegen eine Zugriffspunkt-, Bucket- oder Konto-Einstellung verstoßen.

Die vier von Amazon S3 Block Public Access bereitgestellten Einstellungen lauten wie folgt:

- `BlockPublicAcls`— Diese Einstellung gilt für `PutBucketAcl`, `PutObjectAcl`, `PutObject`, `CreateBucketCopyObject`, und `POST Object` Anfragen. Die `BlockPublicAcls` Einstellung verursacht das folgende Verhalten:
 - `PutBucketAcl` und `PutObjectAcl` Aufrufe schlagen fehl, wenn die angegebene Zugriffskontrollliste (ACL) öffentlich ist.
 - `PutObject` Aufrufe schlagen fehl, wenn die Anfrage ein öffentliches Objekt beinhaltet ACL.
 - Wenn diese Einstellung auf ein Konto angewendet wird, schlagen `CreateBucket` Aufrufe mit einer HTTP 400 (Bad Request) -Antwort fehl, wenn die Anfrage ein öffentliches Objekt beinhaltet ACL.

Wenn beispielsweise der Zugriff auf eine CopyObject Anfrage aufgrund der BlockPublicAcls Einstellung verweigert wird, erhalten Sie die folgende Meldung:

```
An error occurred (AccessDenied) when calling the CopyObject operation:  
User: arn:aws:sts::123456789012:user/MaryMajor is not authorized to  
perform: s3:CopyObject on resource: "arn:aws:s3::amzn-s3-demo-bucket1/object-name"  
because public access control lists (ACLs) are blocked by the BlockPublicAcls block  
public access setting.
```

- **IgnorePublicAcls**— Die IgnorePublicAcls Einstellung veranlasst Amazon S3, alle öffentlichen ACLs Informationen in einem Bucket und alle darin enthaltenen Objekte zu ignorieren. Wenn die Genehmigung Ihrer Anfrage nur von einer Öffentlichkeit erteilt wirdACL, lehnt die IgnorePublicAcls Einstellung die Anfrage ab.

Jede Ablehnung, die sich aus der IgnorePublicAcls Einstellung ergibt, ist implizit. Wenn Sie beispielsweise eine GetObject Anfrage aufgrund IgnorePublicAcls eines öffentlichen Zugriffs ablehnenACL, erhalten Sie die folgende Meldung:

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:  
s3:GetObject because no resource-based policy allows the s3:GetObject action
```

- **BlockPublicPolicy**— Diese Einstellung gilt für PutAccessPointPolicy Anfragen PutBucketPolicy und.

Die Einstellung BlockPublicPolicy für einen Bucket führt dazu, dass Amazon S3 Aufrufe ablehnt, PutBucketPolicy wenn die angegebene Bucket-Richtlinie öffentlichen Zugriff zulässt. Diese Einstellung veranlasst Amazon S3 auch, Anrufe an PutAccessPointPolicy alle Access Points des Buckets mit demselben Konto abzulehnen, wenn die angegebene Richtlinie öffentlichen Zugriff zulässt.

Die Einstellung BlockPublicPolicy für einen Access Point veranlasst Amazon S3, Anrufe an PutAccessPointPolicy und über den Access Point getätigte Anrufe abzulehnenPutBucketPolicy, wenn die angegebene Richtlinie (entweder für den Access Point oder den zugrundeliegenden Bucket) den öffentlichen Zugriff zulässt.

Wenn beispielsweise der Zugriff auf eine PutBucketPolicy Anfrage aufgrund der BlockPublicPolicy Einstellung verweigert wird, erhalten Sie die folgende Meldung:

```
An error occurred (AccessDenied) when calling the PutBucketPolicy operation:
```

```
User: arn:aws:sts::123456789012:user/MaryMajor is not authorized to
perform: s3:PutBucketPolicy on resource: "arn:aws:s3::amzn-s3-demo-bucket1/object-
name"
because public policies are blocked by the BlockPublicPolicy block public
access setting.
```

- **RestrictPublicBuckets**— Die **RestrictPublicBuckets** Einstellung beschränkt den Zugriff auf einen Access Point oder Bucket mit einer öffentlichen Richtlinie auf AWS-Service Prinzipale und autorisierte Benutzer innerhalb des Kontos des Bucket-Besitzers und des Accounts des Access Point-Besitzers. Diese Einstellung blockiert jeglichen kontoübergreifenden Zugriff auf den Access Point oder Bucket (außer von AWS-Service Principals), wobei Benutzer innerhalb des Kontos weiterhin den Access Point oder Bucket verwalten können. Diese Einstellung weist auch alle anonymen (oder unsignierten) Aufrufe zurück.

Jede Ablehnung, die sich aus der **RestrictPublicBuckets** Einstellung ergibt, ist explizit. Wenn beispielsweise eine **GetObject** Anfrage aufgrund einer **RestrictPublicBuckets** öffentlichen Bucket- oder Access-Point-Richtlinie abgelehnt wird, erhalten Sie die folgende Meldung:

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a resource-based policy
```

Weitere Informationen zu diesen Einstellungen finden Sie unter [the section called “Block Public Access-Einstellungen”](#). Informationen zum Überprüfen und Aktualisieren dieser Einstellungen finden Sie unter [the section called “Verwenden von Block Public Access”](#).

Bucket-Richtlinien und IAM Richtlinien

Vorgänge auf Bucket-Ebene

Wenn keine Bucket-Richtlinie vorhanden ist, erlaubt der Bucket implizit Anfragen von beliebigen AWS Identity and Access Management (IAM) Identität im Konto des Bucket-Besitzers. Der Bucket lehnt auch implizit Anfragen von anderen IAM Identitäten von anderen Konten sowie anonyme (unsignierte) Anfragen ab. Wenn jedoch keine IAM Benutzerrichtlinie vorhanden ist, ist der Anforderer (es sei denn, es handelt sich um AWS-Konto Root-Benutzer) wird implizit daran gehindert, Anfragen zu stellen. Weitere Informationen zu dieser Bewertungslogik finden Sie im IAMBenutzerhandbuch unter [Feststellen, ob eine Anfrage innerhalb eines Kontos verweigert oder zulässig ist](#).

Vorgänge auf Objektebene

Wenn das Objekt dem Konto gehört, das den Bucket besitzt, funktionieren die Bucket-Richtlinie und die IAM Benutzerrichtlinie für Operationen auf Objektebene genauso wie für Operationen auf Bucket-Ebene. Wenn beispielsweise keine Bucket-Richtlinie vorhanden ist, lässt der Bucket implizit Objektanforderungen von beliebigen Identitäten im Konto des Bucket-Besitzers zu. IAM Der Bucket lehnt auch implizit Objektanfragen von anderen IAM Identitäten aus anderen Konten sowie anonyme (unsigned) Anfragen ab. Wenn jedoch keine IAM Benutzerrichtlinie vorhanden ist, ist der Anforderer (es sei denn, es handelt sich um AWS-Konto Root-Benutzer) wird implizit daran gehindert, Objektanforderungen zu stellen.

Wenn das Objekt einem externen Konto gehört, kann der Zugriff auf das Objekt nur über Objektzugriffskontrolllisten (ACLs) gewährt werden. Die Bucket-Richtlinie und die IAM Benutzerrichtlinie können weiterhin verwendet werden, um Objektanforderungen abzulehnen.

Um sicherzustellen, dass Ihre Bucket-Richtlinie oder IAM Benutzerrichtlinie nicht den Fehler Access Denied (403 Forbidden) verursacht, stellen Sie daher sicher, dass die folgenden Anforderungen erfüllt sind:

- Für den Zugriff auf dasselbe Konto darf es weder in der Bucket-Richtlinie noch in der Benutzerrichtlinie eine ausdrückliche Deny Erklärung gegen den Antragsteller geben, dem Sie Berechtigungen gewähren möchten. IAM Wenn Sie Berechtigungen nur mithilfe der Bucket-Richtlinie und der IAM Benutzerrichtlinie gewähren möchten, muss in einer dieser Richtlinien mindestens eine ausdrückliche Allow Aussage enthalten sein.
- Für kontoübergreifenden Zugriff darf es weder in der Bucket-Richtlinie noch in der IAM Benutzerrichtlinie eine ausdrückliche Deny Erklärung gegen den Antragsteller geben, dem Sie Berechtigungen gewähren möchten. Wenn Sie kontoübergreifende Berechtigungen nur mithilfe der Bucket-Richtlinie und der IAM Benutzerrichtlinie gewähren möchten, stellen Sie sicher, dass sowohl die Bucket-Richtlinie als auch die IAM Benutzerrichtlinie des Anforderers eine ausdrückliche Erklärung enthalten. Allow

Note

Allow-Anweisungen in einer Bucket-Richtlinie gelten nur für Objekte, [die demselben Konto gehören, das im Besitz des Buckets ist](#). Deny-Anweisungen in einer Bucket-Richtlinie gelten jedoch für alle Objekte, unabhängig von der Objekteigentümerschaft.

So überprüfen oder bearbeiten Sie Ihre Bucket-Richtlinie

Note

Um eine Bucket-Richtlinie anzuzeigen oder zu bearbeiten, benötigen Sie die Berechtigung `s3:GetBucketPolicy`.

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Bucket-Richtlinie anzeigen oder bearbeiten möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie unter Bucket-Richtlinie Bearbeiten aus. Die Seite Bucket-Richtlinie bearbeiten wird angezeigt.

Um Ihre Bucket-Richtlinie zu überprüfen oder zu bearbeiten, verwenden Sie AWS Command Line Interface (AWS CLI), verwenden Sie [get-bucket-policy](#) Befehl.

Note

Wenn Sie aufgrund einer falschen Bucket-Richtlinie aus einem Bucket ausgesperrt werden, [melden Sie sich bei AWS Management Console indem Sie Ihr AWS-Konto Root-Benutzeranmeldedaten](#). Um wieder Zugriff auf Ihren Bucket zu erhalten, stellen Sie sicher, dass Sie die falsche Bucket-Richtlinie löschen, indem Sie Ihre AWS-Konto Root-Benutzeranmeldedaten.

Tipps zum Überprüfen von Berechtigungen

Gehen Sie wie folgt vor, um zu überprüfen, ob der Anforderer über die erforderlichen Berechtigungen für die Ausführung einer Amazon-S3-Operation verfügt:

- Ermitteln Sie den Anforderer. Wenn es sich um eine unsignierte Anfrage handelt, handelt es sich um eine anonyme Anfrage ohne IAM Benutzerrichtlinie. Wenn es sich um eine Anfrage handelt, für

die eine vorsignierte Anfrage verwendet wird URL, gilt dieselbe Benutzerrichtlinie wie für den IAM Benutzer oder die Rolle, die die Anfrage signiert hat.

- Stellen Sie sicher, dass Sie den richtigen IAM Benutzer oder die richtige Rolle verwenden. Sie können Ihren IAM Benutzer oder Ihre Rolle verifizieren, indem Sie in der oberen rechten Ecke des AWS Management Console oder indem Sie den [aws sts get-caller-identity](#) Befehl.
- Überprüfen Sie die IAM Richtlinien, die sich auf den IAM Benutzer oder die Rolle beziehen. Sie können eine der folgenden Methoden verwenden:
 - [Testen Sie IAM Richtlinien mit dem IAM Richtliniensimulator](#).
 - Überprüfen Sie die verschiedenen [IAM Richtlinientypen](#).
 - [Bearbeiten Sie bei Bedarf Ihre IAM Benutzerrichtlinie](#).
- Sehen Sie sich die folgenden Beispiele für Richtlinien an, die den Zugriff explizit verweigern oder zulassen:
 - IAM Benutzerrichtlinie „Explizite Zulassung“ [IAM: Erlaubt und verweigert programmgesteuert und in der Konsole den Zugriff auf mehrere Dienste](#)
 - Explizite Richtlinie „Bucket zulassen“: [Erteilt mehreren Konten die Erlaubnis, Objekte hochzuladen oder Objekte öffentlich zugänglich zu machen ACLs](#)
 - Richtlinie „IAM Benutzer ausdrücklich ablehnen“: [AWS: Verweigert den Zugriff auf AWS basierend auf dem angeforderten AWS-Region](#)
 - Explizite Richtlinie zum Ablehnen von Buckets: [SSE erforderlich — KMS für alle Objekte, die in einen Bucket geschrieben werden](#)

Amazon S3 ACL S3-Einstellungen

Überprüfen Sie bei der Überprüfung Ihrer ACL Einstellungen zunächst [Ihre Einstellungen für Object Ownership](#), um zu überprüfen, ob ACLs für den Bucket aktiviert sind. Beachten Sie, dass ACL Berechtigungen nur zur Erteilung von Berechtigungen und nicht zur Ablehnung von Anfragen verwendet werden können. ACLs kann auch nicht verwendet werden, um Anfordern Zugriff zu gewähren, die durch ausdrückliche Ablehnungen in Bucket- oder IAM Benutzerrichtlinien abgelehnt wurden.

Die Einstellung für Object Ownership ist auf „Bucket-Eigentümer erzwungen“ gesetzt.

Wenn die Einstellung „Bucket Owner enforced“ aktiviert ist, ist es unwahrscheinlich, dass die ACL Einstellungen zu einem Fehler „Zugriff verweigert“ (403 Forbidden) führen, da diese Einstellung


alles deaktiviert ACLs, was für Buckets und Objekte gilt. „Bucket-Eigentümer erzwungen“ ist die Standardeinstellung (und empfohlene Einstellung) für Amazon-S3-Buckets.

Die Einstellung für Object Ownership ist auf „Bucket-Eigentümer bevorzugt“ oder „Objektschreiber“ gesetzt.

ACL Mit der bevorzugten Einstellung des Bucket-Besitzers oder der Einstellung Object Writer sind die Berechtigungen weiterhin gültig. Es gibt zwei Arten von ACLs: Bucket ACLs und Objekt ACLs. Die Unterschiede zwischen diesen beiden Typen finden Sie unter [Zuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen](#). ACLs


[Überprüfen Sie je nach der Aktion der abgelehnten Anfrage die ACL Berechtigungen für Ihren Bucket oder das Objekt:](#)

- Wenn Amazon S3 ein LIST PUT Objekt oder eine PutBucketAc1 Anfrage abgelehnt hat, [überprüfen Sie die ACL Berechtigungen für Ihren Bucket](#). GetBucketAc1

 Note

Mit den ACL Bucket-Einstellungen können Sie keine GET Objektberechtigungen gewähren.

- Wenn Amazon S3 eine GET Anfrage für ein S3-Objekt abgelehnt hat, oder [PutObjectAc1](#)Anfrage und [überprüfen Sie anschließend die ACL Berechtigungen für das Objekt](#).

 Important

Wenn es sich bei dem Konto, dem das Objekt gehört, nicht um das Konto handelt, das im Besitz des Buckets ist, wird der Zugriff auf das Objekt nicht durch die Bucket-Richtlinie gesteuert.

Beheben eines Fehlers aufgrund einer Zugriffsverweigerung (403 Forbidden) bei einer **GET**-Objekt-Anforderung während einer kontoübergreifenden Objekteigentümerschaft

Überprüfen Sie die [Einstellungen für Object Ownership](#) des Buckets, um den Objekteigentümer zu ermitteln. Wenn Sie Zugriff auf das [Objekt](#) haben ACLs, können Sie auch das Konto des Objektbesitzers überprüfen. (Um das Konto des Objekteigentümers einzusehen, überprüfen Sie die ACL Objekteinstellungen in der Amazon S3 S3-Konsole.) Alternativ können Sie auch eine GetObjectAc1-Anforderung stellen, um die [kanonische ID](#) des Objekteigentümers zu ermitteln

und so das Konto des Objekteigentümers überprüfen zu können. ACLs erteilen Sie dem Konto des Objekteigentümers standardmäßig explizite Genehmigungsberechtigungen für GET-Anfragen.

Nachdem Sie bestätigt haben, dass der Objekteigentümer nicht mit dem Bucket-Eigentümer identisch ist, wählen Sie je nach Anwendungsfall und Zugriffsebene eine der folgenden Methoden aus, um den Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) zu beheben:

- Deaktivieren ACLs (empfohlen) — Diese Methode gilt für alle Objekte und kann vom Bucket-Besitzer ausgeführt werden. Bei dieser Methode erhält der Bucket-Eigentümer automatisch das Eigentum an jedem Objekt im Bucket und die volle Kontrolle darüber. Bevor Sie diese Methode implementieren, überprüfen Sie die [Voraussetzungen für die Deaktivierung ACLs](#). Informationen dazu, wie Sie Ihren Bucket auf den Modus „Bucket-Eigentümer erzwungen“ (empfohlen) setzen, finden Sie unter [Einstellung für Object Ownership für einen vorhandenen Bucket](#).

⚠ Important

Um zu verhindern, dass der Fehler „Zugriff verweigert“ (403 Forbidden) auftritt, sollten Sie vor der Deaktivierung ACLs die ACL-Berechtigungen auf eine Bucket-Richtlinie migrieren. Weitere Informationen finden Sie unter [Beispiele für Bucket-Richtlinien für die Migration von ACL-Berechtigungen](#).

- Objekteigentümer in Bucket-Eigentümer ändern – Diese Methode kann auf einzelne Objekte angewendet werden, doch nur der Objekteigentümer (oder ein Benutzer mit den entsprechenden Berechtigungen) kann die Eigentümerschaft eines Objekts ändern. Zusätzliche PUT-Kosten können anfallen. (Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).) Diese Methode gewährt dem Bucket-Eigentümer das vollständige Eigentum an dem Objekt, sodass der Bucket-Eigentümer den Zugriff auf das Objekt über eine Bucket-Richtlinie steuern kann.

Führen Sie einen der folgenden Schritte aus, um die Objekteigentümerschaft zu ändern:

- Sie (der Bucket-Eigentümer) können [das Objekt wieder in den Bucket kopieren](#).
- Sie können die Einstellung für Object Ownership des Buckets auf „Bucket-Eigentümer bevorzugt“ setzen. Wenn die Versionsverwaltung deaktiviert ist, werden die Objekte im Bucket überschrieben. Wenn die Versionsverwaltung aktiviert ist, werden doppelte Versionen desselben Objekts im Bucket angezeigt, für die der Bucket-Eigentümer [eine Lebenszyklusregel festlegen kann, damit sie ablaufen](#). Anweisungen zum Ändern der Einstellung für Object Ownership finden Sie unter [Einstellung für Object Ownership für einen vorhandenen Bucket](#).

Note

Wenn Sie Ihre Einstellung für Object Ownership in „Bucket-Eigentümer bevorzugt“ aktualisieren, wird die Einstellung nur auf neue Objekte angewendet, die in den Bucket hochgeladen werden.

- Sie können den Objekteigentümer das Objekt zusammen mit dem gespeicherten Objekt `bucket-owner-full-control` erneut hochladen lassenACL.

Note

Für kontoübergreifende Uploads können Sie das `bucket-owner-full-control` gespeicherte Objekt auch ACL in Ihrer Bucket-Richtlinie vorschreiben. Ein Beispiel für eine Bucket-Richtlinie finden Sie unter [Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt](#).

- Objektschreiber als Objekteigentümer beibehalten – Mit dieser Methode wird der Objekteigentümer nicht geändert, Sie können jedoch den Zugriff auf Objekte einzeln gewähren. Um Zugriff auf ein Objekt zu gewähren, müssen Sie über die Berechtigung `PutObjectAcl` für das Objekt verfügen. Um dann den Fehler „Zugriff verweigert“ (403 Forbidden) zu beheben, fügen Sie den Antragsteller als Empfänger für den Zugriff [auf](#) das Objekt im Objekt hinzu. ACLs Weitere Informationen finden Sie unter [Konfiguration ACLs](#).

S3–Block-Public-Access-Einstellungen

Wenn die fehlgeschlagene Anfrage öffentlichen Zugriff oder öffentliche Richtlinien beinhaltet, überprüfen Sie die S3-Einstellungen für Block Public Access in Ihrem Konto, Bucket oder Access Point. Weitere Informationen zur Behebung von Fehlern im Zusammenhang mit der Einstellung „Zugriff verweigert“ im Zusammenhang mit den Einstellungen zum Blockieren des öffentlichen Zugriffs in S3 finden Sie unter [the section called “Der Zugriff wurde aufgrund der Einstellungen zum Blockieren des öffentlichen Zugriffs verweigert”](#).

Amazon-S3-Verschlüsselungseinstellungen

Amazon S3 unterstützt die serverseitige Verschlüsselung in Ihrem Bucket. Serverseitige Verschlüsselung ist die Verschlüsselung von Daten am Zielort durch die Anwendung oder den

Service, der sie erhält. Amazon S3 verschlüsselt Ihre Daten auf Objektebene beim Schreiben auf Festplatten in AWS Rechenzentren und entschlüsselt sie für Sie, wenn Sie darauf zugreifen.

Standardmäßig wendet Amazon S3 jetzt serverseitige Verschlüsselung mit verwalteten Amazon S3-Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Amazon S3 ermöglicht es Ihnen auch, die serverseitige Verschlüsselungsmethode beim Hochladen von Objekten anzugeben.

So überprüfen Sie den Status der serverseitigen Verschlüsselung und die Verschlüsselungseinstellungen Ihres Buckets

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket aus, für den Sie die Verschlüsselungseinstellungen überprüfen möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie nach unten zum Abschnitt Standardverschlüsselung und sehen Sie sich die Einstellungen für den Verschlüsselungstyp an.

Um Ihre Verschlüsselungseinstellungen zu überprüfen, verwenden Sie AWS CLI, verwenden Sie [get-bucket-encryption](#) Befehl.

So überprüfen Sie den Verschlüsselungsstatus eines Objekts

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
4. Wählen Sie in der Liste Objekte den Namen des Objekts aus, für das Sie eine Verschlüsselung hinzufügen oder ändern möchten.

Die Seite mit den Objektdetails wird angezeigt.

5. Scrollen Sie nach unten zum Abschnitt Serverseitige Verschlüsselungseinstellungen, um die serverseitigen Verschlüsselungseinstellungen des Objekts anzuzeigen.

Um den Verschlüsselungsstatus Ihres Objekts zu überprüfen, verwenden Sie AWS CLI, verwenden Sie [head-object](#) Befehl.

Verschlüsselungs- und Berechtigungsanforderungen

Amazon S3 unterstützt drei Arten von serverseitiger Verschlüsselung:

- Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)
- Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS)
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Stellen Sie unter Berücksichtigung Ihrer Verschlüsselungseinstellungen sicher, dass die folgenden Berechtigungen erfüllt sind:

- SSE-S3 — Es sind keine zusätzlichen Berechtigungen erforderlich.
- SSE-KMS (mit einem vom Kunden verwalteten Schlüssel) — Um Objekte hochzuladen, ist die `kms:GenerateDataKey` Erlaubnis für AWS KMS key ist erforderlich. Um Objekte herunterzuladen und mehrteilige Uploads von Objekten durchzuführen, ist die `kms:Decrypt` Erlaubnis für den KMS Schlüssel erforderlich.
- SSE-KMS (mit einem Von AWS verwalteter Schlüssel) — Der Anforderer muss aus demselben Konto stammen, dem der `aws/s3` KMS Schlüssel gehört. Der Anforderer muss außerdem über die richtigen Amazon-S3-Berechtigungen verfügen, um auf das Objekt zugreifen zu können.
- SSE-C (mit einem vom Kunden bereitgestellten Schlüssel) — Es sind keine zusätzlichen Berechtigungen erforderlich. Sie können die Bucket-Richtlinie so konfigurieren, dass [eine serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln für Objekte in Ihrem Bucket erforderlich und beschränkt ist](#).

Wenn das Objekt mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, stellen Sie sicher, dass die KMS Schlüsselrichtlinie es Ihnen ermöglicht, die `kms:GenerateDataKey` `kms:Decrypt` OR-Aktionen auszuführen. Anweisungen zur Überprüfung Ihrer KMS Schlüsselrichtlinie finden Sie unter [Eine wichtige Richtlinie anzeigen](#) im AWS Key Management Service Leitfaden für Entwickler.

S3-Einstellungen für die Objektsperre

Wenn in Ihrem Bucket die [S3-Objektsperre](#) aktiviert ist und das Objekt durch einen [Aufbewahrungszeitraum](#) oder eine [rechtliche Aufbewahrungspflicht](#) geschützt ist, gibt Amazon

S3 beim Versuch, das Objekt zu löschen, einen Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) zurück.

So überprüfen Sie, ob für den Bucket eine Objektsperre aktiviert ist

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie überprüfen möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie nach unten zum Abschnitt Objektsperre. Überprüfen Sie, ob die Einstellung für Objektsperre Aktiviert oder Deaktiviert lautet.

Um festzustellen, ob das Objekt durch einen Aufbewahrungszeitraum oder eine rechtliche Aufbewahrungspflicht geschützt ist, [zeigen Sie die Sperrinformationen](#) für Ihr Objekt an.

Wenn das Objekt durch einen Aufbewahrungszeitraum oder eine rechtliche Aufbewahrungspflicht geschützt ist, überprüfen Sie Folgendes:

- Wenn die Objektversion durch den Compliance-Aufbewahrungsmodus geschützt ist, gibt es keine Möglichkeit, sie dauerhaft zu löschen. Eine permanente DELETE-Anfrage von einem beliebigen Antragsteller, einschließlich AWS-Konto Root-Benutzer, führt zu einem Fehler „Zugriff verweigert“ (403 Forbidden). Beachten Sie außerdem, dass Amazon S3 eine [Löschmarkierung](#) für das Objekt erstellt, wenn Sie eine DELETE-Anforderung für ein Objekt einreichen, das durch den Compliance-Aufbewahrungsmodus geschützt ist.
- Wenn die Objektversion durch den Governance-Aufbewahrungsmodus geschützt ist und Sie über die Berechtigung `s3:BypassGovernanceRetention` verfügen, können Sie den Schutz umgehen und die Version dauerhaft löschen. Weitere Informationen finden Sie unter [Umgehen des Governance-Modus](#).
- Wenn die Objektversion durch eine rechtliche Aufbewahrungspflicht geschützt ist, kann eine permanente DELETE-Anforderung zu einem Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) führen. Um die Objektversion dauerhaft zu löschen, müssen Sie die rechtliche Aufbewahrungspflicht für die Objektversion aufheben. Um eine rechtliche Aufbewahrungspflicht aufzuheben, benötigen Sie die Berechtigung `s3:PutObjectLegalHold`. Weitere Informationen zum Aufheben einer rechtlichen Aufbewahrungspflicht finden Sie unter [Konfigurieren von S3 Object Lock](#).

VPC Endpunktrichtlinien

Wenn Sie über einen Virtual Private Cloud (VPC) -Endpunkt auf Amazon S3 zugreifen, stellen Sie sicher, dass die VPC Endpunktrichtlinie Sie nicht am Zugriff auf Ihre Amazon S3 S3-Ressourcen hindert. Standardmäßig erlaubt die VPC Endpunktrichtlinie alle Anfragen an Amazon S3. Sie können die VPC Endpunktrichtlinie auch so konfigurieren, dass bestimmte Anfragen eingeschränkt werden. Informationen dazu, wie Sie Ihre VPC Endpunktrichtlinie überprüfen können, finden Sie in den folgenden Ressourcen:

- [the section called “Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — implizite Ablehnung”](#)
- [the section called “Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — ausdrückliche Ablehnung”](#)
- [Steuern Sie den Zugriff auf VPC Endgeräte mithilfe von Endpunktrichtlinien](#) in AWS PrivateLink Leitfaden

AWS Organizations policies

Wenn Ihre AWS-Konto gehört einer Organisation an, AWS Organizations Richtlinien können Sie daran hindern, auf Amazon S3 S3-Ressourcen zuzugreifen. Standardmäßig AWS Organizations Richtlinien blockieren keine Anfragen an Amazon S3. Stellen Sie jedoch sicher, dass Ihre AWS Organizations Richtlinien wurden nicht so konfiguriert, dass sie den Zugriff auf S3-Buckets blockieren. Für Anweisungen, wie Sie Ihre überprüfen können AWS Organizations Richtlinien finden Sie in den folgenden Ressourcen:

- [the section called “Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – implizite Verweigerung”](#)
- [the section called “Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – explizite Verweigerung”](#)
- [Auflistung aller Richtlinien](#) in der AWS Organizations Benutzerleitfaden

Zugriffspunkteinstellungen

Wenn Sie bei Anforderungen über Amazon-S3-Zugriffspunkte einen Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) erhalten, müssen Sie möglicherweise Folgendes überprüfen:

- Die Konfigurationen Ihrer Zugriffspunkte

- Die IAM Benutzerrichtlinie, die für Ihre Access Points verwendet wird
- Die Bucket-Richtlinie, die zur Verwaltung oder Konfiguration Ihrer kontoübergreifenden Zugriffspunkte verwendet wird

Zugriffspunktkonfigurationen und -richtlinien

- Wenn Sie einen Zugriffspunkt erstellen, können Sie wählen, ob Sie das Internet oder VPC als Netzwerkquelle angeben möchten. Wenn der Netzwerkursprung auf VPC Nur gesetzt ist, lehnt Amazon S3 alle Anfragen an den Access Point ab, die nicht von dem angegebenen stammen VPC. Informationen zum Überprüfen des Netzwerkursprungs Ihres Zugriffspunkts finden Sie unter [Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind](#).
- Mit Zugriffspunkten können Sie auch benutzerdefinierte Block-Public-Access-Einstellungen konfigurieren, die ähnlich wie die Block-Public-Access-Einstellungen auf Bucket- oder Kontoebene funktionieren. Informationen zu Ihren benutzerdefinierten Block-Public-Access-Einstellungen finden Sie unter [Verwalten des öffentlichen Zugriffs auf Zugriffspunkte](#).
- Um mithilfe von Access Points erfolgreiche Anfragen an Amazon S3 zu stellen, stellen Sie sicher, dass der Anforderer über die erforderlichen IAM Berechtigungen verfügt. Weitere Informationen finden Sie unter [Konfiguration von IAM Richtlinien für die Verwendung von Access Points](#).
- Wenn die Anforderung kontoübergreifende Zugriffspunkte umfasst, stellen Sie sicher, dass der Bucket-Eigentümer die Bucket-Richtlinie aktualisiert hat, um Anforderungen vom Zugriffspunkt zu autorisieren. Weitere Informationen finden Sie unter [Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte](#).

Wenn der Fehler Zugriff verweigert (403 Forbidden) weiterhin besteht, nachdem Sie alle Artikel in diesem Thema überprüft haben, [rufen Sie Ihre Amazon S3-Anforderungs-ID ab und kontaktieren Sie](#) AWS Support für zusätzliche Hinweise.

Fehlerbehebung bei Batch Operations

In den folgenden Themen werden häufige Fehler behandelt, um Sie dabei zu unterstützen, Probleme bei der Arbeit mit Batch Operations zu beheben.

Häufige Fehler

- [Der Auftragsbericht wird nicht bereitgestellt, wenn ein Problem mit Berechtigungen besteht oder ein Aufbewahrungsmodus der S3-Objektsperre aktiviert ist.](#)

- [S3-Batchreplikation mit folgendem Fehler fehlgeschlagen: Bei der Manifestgenerierung wurden keine Schlüssel gefunden, die den Filterkriterien entsprechen](#)
- [Fehler bei Batch Operations nach dem Hinzufügen einer neuen Replikationsregel zu einer vorhandenen Replikationskonfiguration](#)
- [Bei Batch-Vorgängen scheitern Objekte mit dem Fehler 400 InvalidRequest: Aufgabe ist aufgrund fehlender Objekte fehlgeschlagen VersionId](#)
- [Fehler bei Auftragserstellung mit aktivierter Auftrags-Tag-Option](#)
- [Zugriff zum Lesen des Manifests verweigert](#)

Der Auftragsbericht wird nicht bereitgestellt, wenn ein Problem mit Berechtigungen besteht oder ein Aufbewahrungsmodus der S3-Objektsperre aktiviert ist.

Der folgende Fehler tritt auf, wenn erforderliche Berechtigungen fehlen oder ein Aufbewahrungsmodus der Objektsperre (entweder Governance-Modus oder Compliance-Modus) für den Ziel-Bucket aktiviert ist.

Fehler: Fehlerursachen. Der Auftragsbericht konnte nicht in Ihren Berichts-Bucket geschrieben werden. Überprüfen Sie Ihre Berechtigungen.

Die IAM Rollen- und Vertrauensrichtlinie müssen so konfiguriert werden, dass S3 Batch Operations Zugriff auf PUT Objekte in dem Bucket gewährt, in den der Bericht geliefert wird. Wenn diese erforderlichen Berechtigungen fehlen, schlägt die Bereitstellung des Auftragsberichts fehl.

Wenn ein Aufbewahrungsmodus aktiviert ist, ist der Bucket geschützt write-once-read-many (WORM). Die Objektsperre mit dem für den Ziel-Bucket aktivierten Aufbewahrungsmodus wird nicht unterstützt. Daher schlägt die Bereitstellung des Auftragsabschlussberichts fehl. Wählen Sie zum Beheben dieses Problems einen Ziel-Bucket für Ihre Abschlussberichte zu Aufträgen aus, für den kein Aufbewahrungsmodus der Objektsperre aktiviert ist.

S3-Batchreplikation mit folgendem Fehler fehlgeschlagen: Bei der Manifestgenerierung wurden keine Schlüssel gefunden, die den Filterkriterien entsprechen

Fehler: Bei der Manifestgenerierung wurden keine Schlüssel gefunden, die den Filterkriterien entsprechen.

Dieser Fehler tritt aus einem der folgenden Gründe auf:

- Wenn Objekte im Quell-Bucket in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert werden.

Um die Batchreplikation für diese Objekte zu verwenden, stellen Sie sie zunächst in der Speicherkategorie S3 Standard wieder her, indem Sie die Operation S3 Initiate Restore Object in einem Batch-Operations-Auftrag verwenden. Weitere Informationen finden Sie unter [Wiederherstellen eines archivierten Objekts](#) und [Wiederherstellen von Objekten \(Batch Operations\)](#). Nachdem Sie die Objekte wiederhergestellt haben, können Sie sie mithilfe eines Batchreplikationsauftrags replizieren.

- Wenn die angegebenen Filterkriterien mit keinen gültigen Objekten im Quell-Bucket übereinstimmen.

Überprüfen und korrigieren Sie die Filterkriterien. In der Regel für die Batch-Replikation suchen die Filterkriterien beispielsweise nach allen Objekten in der *amzn-s3-demo-bucket* mit dem Präfix `Tax/`. Wenn der Präfixname falsch eingegeben wurde, mit einem Schrägstrich am Anfang und am Ende `/Tax/` statt nur am Ende, wurden keine S3-Objekte gefunden. Um den Fehler zu beheben, korrigieren Sie das Präfix, in diesem Fall von `/Tax/` bis `Tax/` in der Replikationsregel.

Fehler bei Batch Operations nach dem Hinzufügen einer neuen Replikationsregel zu einer vorhandenen Replikationskonfiguration

Batch Operations versucht, eine vorhandene Objektreplikation für jede Regel in der Replikationskonfiguration des Quell-Buckets auszuführen. Wenn es Probleme mit einer der vorhandenen Replikationsregeln gibt, können Fehler auftreten.

Im Abschlussbericht des Batch-Operations-Auftrags werden die Ursachen für das Fehlschlagen des Auftrags erläutert. Eine Liste mit häufig auftretenden Fehlern finden Sie unter [Gründe für das Fehlschlagen der Replikation in Amazon S3](#).

Bei Batch-Vorgängen scheitern Objekte mit dem Fehler 400 InvalidRequest: Aufgabe ist aufgrund fehlender Objekte fehlgeschlagen VersionId

Der folgende Beispielfehler tritt auf, wenn ein Batch-Operations-Auftrag Aktionen für Objekte in einem versionsfähigen Bucket ausführt und im Manifest ein Objekt mit leerem Versions-ID-Feld feststellt.

Fehler: *BUCKET_NAME, prefix/file_name*, fehlgeschlagen,400,, Die Aufgabe ist fehlgeschlagenInvalidRequest, weil sie fehlt VersionId

Dieser Fehler tritt auf, weil das Versions-ID-Feld im Manifest eine leere Zeichenfolge und nicht die Literalzeichenfolge null ist.

Batch Operations schlägt für diese(s) spezielle(n) Objekt(e) fehl, aber nicht für den gesamten Auftrag. Dieses Problem tritt auf, wenn das Manifestformat so konfiguriert ist, dass die Version IDs während des Vorgangs verwendet wird. Bei Aufträgen ohne Versionierung tritt dieses Problem nicht auf, da sie nur mit der neuesten Version jedes Objekts ausgeführt werden und die Version IDs im Manifest ignoriert wird.

Um dieses Problem zu beheben, konvertieren Sie die leere Version IDs in Zeichenketten. null
Weitere Informationen finden Sie unter [the section called "Konvertieren leerer Versions-ID-Zeichenfolgen in Null-Zeichenfolgen"](#).

Fehler bei Auftragserstellung mit aktivierter Auftrags-Tag-Option

Ohne die Berechtigung `s3:PutJobTagging` führt das Erstellen von Batch-Operations-Aufträgen mit aktivierter Auftrags-Tag-Option zum Fehler 403 `access denied`.

Um Batch Operations-Jobs mit aktivierter Job-Tag-Option zu erstellen, muss der Benutzer AWS Identity and Access Management (IAM), der den Batch Operations-Job erstellt, zusätzlich zur `s3:PutJobTagging` Berechtigung auch über die `s3:CreateJob` entsprechende Berechtigung verfügen.

Weitere Informationen zu den erforderlichen Berechtigungen für Batch Operations finden Sie unter [the section called "Gewähren von Berechtigungen"](#).

Zugriff zum Lesen des Manifests verweigert

Wenn Sie versuchen, einen Batch-Operations-Auftrag zu erstellen und Batch Operations die Manifestdatei nicht lesen kann, können die folgenden Fehler auftreten.

AWS CLI

Grund des Fehlers Das Manifest zu lesen ist verboten: AccessDenied

Amazon S3-Konsole

Warnung: Die des Manifest-Objekts konnten nicht abgerufen ETag werden. Geben Sie ein anderes Objekt an, um fortzufahren.

Führen Sie die folgenden Schritte aus, um dieses Problem zu beheben:

- Stellen Sie sicher, dass die IAM Rolle für die AWS-Konto , mit der Sie den Batch Operations-Job erstellt haben, über `s3:GetObject` Berechtigungen verfügt. Die IAM Rolle des Kontos muss über `s3:GetObject` Berechtigungen verfügen, damit Batch Operations die Manifestdatei lesen kann.

Weitere Informationen zu den erforderlichen Berechtigungen für Batch Operations finden Sie unter [the section called “Gewähren von Berechtigungen”](#).

- Überprüfen Sie die Metadaten der Manifestobjekte auf etwaige Zugriffskonflikte mit S3 Object Ownership. Informationen zu S3 Object Ownership finden Sie unter [the section called “Steuern der Objekteigentümerschaft”](#).
- Prüfen Sie, ob AWS Key Management Service (AWS KMS) -Schlüssel zum Verschlüsseln der Manifestdatei verwendet werden.

Batch Operations unterstützt CSVInventarberichte, die AWS KMS verschlüsselt sind. Batch Operations unterstützen jedoch keine CSVManifestdateien, die AWS KMS-verschlüsselt sind.

Weitere Informationen finden Sie unter [Konfigurieren von Amazon S3 Inventory](#) und [Angeben eines Manifests](#).

Behebung von Problemen mit Amazon S3 Lifecycle

Die folgenden Informationen können Ihnen helfen, häufiger auftretende Probleme mit den Lebenszyklusregeln von Amazon S3 zu beheben.

Themen

- [Ich habe eine Auflistungsoperation für meinen Bucket ausgeführt und es wurden Objekte angezeigt, von denen ich dachte, dass sie abgelaufen oder aufgrund einer Lebenszyklusregel übergeben worden waren.](#)
- [Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?](#)
- [Die Anzahl meiner S3-Objekte steigt immer noch, auch wenn ich Lebenszyklusregeln für einen Bucket mit aktivierter Versionierung eingerichtet habe.](#)
- [Wie leere ich meinen S3-Bucket mithilfe von Lebenszyklusregeln?](#)

- [Meine Abrechnung für Amazon S3 weist nach der Übergabe von Objekten in eine kostengünstigere Speicherklasse höhere Kosten auf.](#)
- [Ich habe meine Bucket-Richtlinie aktualisiert, meine S3-Objekte werden jedoch noch immer aufgrund abgelaufener Lebenszyklusregeln gelöscht.](#)
- [Kann ich S3-Objekte wiederherstellen, die aufgrund von S3-Lebenszyklusregeln abgelaufen sind?](#)
- [Wie kann ich ein Präfix aus meiner Lebenszyklusregel ausschließen?](#)
- [Wie kann ich mehrere Präfixe in meine Lebenszyklusregel aufnehmen?](#)

Ich habe eine Auflistungsoperation für meinen Bucket ausgeführt und es wurden Objekte angezeigt, von denen ich dachte, dass sie abgelaufen oder aufgrund einer Lebenszyklusregel übergeben worden waren.

[Das Übergeben von Objekten](#) und das [Ablaufen von Objekten](#) in S3-Lebenszyklen sind asynchrone Operationen. Daher kann es zu einer Verzögerung zwischen dem Zeitpunkt kommen, zu dem die Objekte für ein Ablaufen oder eine Übergabe infrage kommen, und dem Zeitpunkt, zu dem sie tatsächlich übergeben werden oder ablaufen. Änderungen an der Abrechnung werden sofort nach Erfüllung der Lebenszyklusregel angewendet, auch wenn die Aktion noch nicht abgeschlossen ist. Eine Ausnahme von diesem Verhalten tritt auf, wenn Sie eine Lebenszyklusregel für die Übergabe in die Speicherklasse S3 Intelligent Tiering festgelegt haben. In diesem Fall treten Abrechnungsänderungen erst auf, wenn das Objekt in S3 Intelligent-Tiering übergeben worden ist. Weitere Informationen zu Änderungen der Abrechnung finden Sie unter [Festlegen der Lebenszyklus-Konfiguration für einen Bucket](#).

Note

Amazon S3 übergibt keine Objekte, die kleiner als 128 KB sind, von der Speicherklasse S3 Standard oder S3 Standard-IA in die Klassen S3 Intelligent-Tiering, S3 Standard-IA oder S3 One Zone-IA.

Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?

Um die im Rahmen von Lebenszyklusregeln durchgeführten Aktionen zu überwachen, können Sie die folgenden Funktionen verwenden:

- S3-Ereignisbenachrichtigungen — Sie können [S3-Ereignisbenachrichtigungen](#) so einrichten, dass Sie über alle Ablaufs- oder Übergangereignisse von S3 Lifecycle informiert werden.
- S3-Serverzugriffsprotokolle — Sie können Serverzugriffsprotokolle für Ihre S3-Buckets aktivieren, um S3-Lifecycle-Aktionen wie Objektübergänge in eine andere Speicherklasse oder Objektlaufzeiten zu erfassen. Weitere Informationen finden Sie unter [Lebenszyklus und Protokollieren](#).

Um die Änderungen an Ihrem Speicher, die durch Lebenszyklusaktionen verursacht werden, täglich zu sehen, empfehlen wir, [S3 Storage Lens-Dashboards](#) zu verwenden, anstatt CloudWatch Amazon-Metriken zu verwenden. In Ihrem Storage Lens-Dashboard können Sie sich die folgenden Metriken ansehen, die die Anzahl oder Größe von Objekten überwachen:

- Aktuelle Versions-Bytes
- Anzahl der Objekte der aktuellen Version
- Nicht aktuelle Versions-Bytes
- Anzahl nicht aktueller Versionsobjekte
- Anzahl der Löschmarkierungsobjekte
- Markierungsspeicherbytes löschen
- Unvollständige Bytes für mehrteilige Uploads
- Anzahl unvollständiger mehrteiliger Uploads

Die Anzahl meiner S3-Objekte steigt immer noch, auch wenn ich Lebenszyklusregeln für einen Bucket mit aktivierter Versionierung eingerichtet habe.

Wenn in einem [Bucket mit aktivierter Versionierung](#) ein Objekt abgelaufen ist, wird das Objekt nicht vollständig aus dem Bucket gelöscht. Stattdessen wird eine [Löschmarkierung](#) als neueste Version des Objekts erstellt. Löschmarkierungen werden weiterhin als Objekte gezählt. Wenn also die Lebenszyklusregel erstellt wird, dass nur die aktuellen Versionen ablaufen, erhöht sich die Anzahl der Objekte im S3-Bucket, anstatt zu sinken.

Nehmen wir beispielsweise an, dass für einen S3-Bucket mit 100 Objekten die Versionsverwaltung aktiviert und die Lebenszyklusregel festgelegt ist, dass aktuelle Versionen des Objekts nach 7 Tagen ablaufen. Nach dem siebten Tag erhöht sich die Anzahl der Objekte auf 200, da zusätzlich

zu den 100 ursprünglichen Objekten, bei denen es sich jetzt um die nicht aktuellen Versionen handelt, 100 Löschmarkierungen erstellt werden. Weitere Informationen zu den Aktionen der S3-Lebenszyklus-Konfigurationsregel für Buckets mit aktivierter Versionsverwaltung finden Sie unter [Festlegen der Lebenszyklus-Konfiguration für einen Bucket](#).

Um Objekte dauerhaft zu entfernen, fügen Sie eine zusätzliche Lebenszykluskonfiguration hinzu, um die vorherigen Versionen der Objekte, abgelaufene Löschmarkierungen und unvollständige mehrteilige Uploads zu löschen. Anweisungen zum Erstellen neuer Lebenszyklusregeln finden Sie unter [Festlegen der Lebenszyklus-Konfiguration für einen Bucket](#).

Note

- Amazon S3 rundet das Übergangs- oder Verfallsdatum eines Objekts auf Mitternacht UTC des nächsten Tages ab.

Bei der Bewertung von Objekten auf Lebenszyklusaktionen verwendet Amazon S3 die Objekterstellungszeit in UTC. Stellen Sie sich zum Beispiel einen Bucket ohne Versionsnummer mit einer Lebenszyklusregel vor, der so konfiguriert ist, dass Objekte nach einem Tag ablaufen. Angenommen, ein Objekt wurde am 1. Januar um 17:05 Uhr PDT (PDT) erstellt, was dem 2. Januar um 00:05 Uhr entspricht. UTC Das Objekt wird UTC am 3. Januar um 00:05 Uhr einen Tag alt, sodass es ablaufen kann, wenn S3 Lifecycle Objekte am 4. Januar um 00:00 UTC Uhr auswertet.

Da Amazon S3 S3-Lebenszyklusaktionen asynchron ausgeführt werden, kann es zu Verzögerungen zwischen dem in der Lebenszyklusregel angegebenen Datum und dem tatsächlichen physischen Übergang des Objekts kommen. Weitere Informationen finden Sie unter [Übergangs- oder Ablaufverzögerung](#).

Weitere Informationen finden Sie unter [Lebenszyklusregeln: Basierend auf dem Alter eines Objekts](#).

- Bei S3-Objekten, die durch die Objektsperre geschützt sind, werden aktuelle Versionen nicht dauerhaft gelöscht. Stattdessen wird den Objekten eine Löschmarkierung hinzugefügt, sodass sie nicht mehr aktuell sind. Nicht aktuelle Versionen werden dann beibehalten und laufen nicht dauerhaft ab.

Wie leere ich meinen S3-Bucket mithilfe von Lebenszyklusregeln?

S3-Lebenszyklusregeln sind ein effektives Tool zum [Leeren eines S3-Buckets](#) mit Millionen von Objekten. Um eine große Anzahl von Objekten aus Ihrem S3-Bucket zu löschen, stellen Sie sicher, dass Sie diese beiden Paare von Lebenszyklusregeln verwenden:

- Aktuelle Objektversionen ablaufen lassen und Vorherige Versionen von Objekten dauerhaft löschen
- Löschmarkierungen für abgelaufenes Objekt löschen und Unvollständige mehrteilige Uploads löschen

Anweisungen zum Erstellen einer neuen Lebenszyklus-Konfigurationsregel finden Sie unter [Festlegen der Lebenszyklus-Konfiguration für einen Bucket](#).

Note

Bei S3-Objekten, die durch die Objektsperre geschützt sind, werden aktuelle Versionen nicht dauerhaft gelöscht. Stattdessen wird den Objekten eine Löschmarkierung hinzugefügt, sodass sie nicht mehr aktuell sind. Nicht aktuelle Versionen werden dann beibehalten und laufen nicht dauerhaft ab.

Meine Abrechnung für Amazon S3 weist nach der Übergabe von Objekten in eine kostengünstigere Speicherklasse höhere Kosten auf.

Es gibt mehrere Gründe, warum Ihre Abrechnung nach der Übergabe von Objekten in eine kostengünstigere Speicherklasse höhere Kosten aufweisen kann:

- S3-Glacier-Overheadgebühren für kleine Objekte

Bei jedem Objekt, das an S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive übergeben wird, ist ein Gesamt-Overhead von 40 KB mit dieser Speicheraktualisierung verbunden. Im Rahmen des Overheads von 40 KB werden 8 KB zum Speichern der Metadaten und des Objektnamens verwendet. Diese 8 KB werden gemäß den S3-Standardgebühren berechnet. Die restlichen 32 KB werden für die Indexierung und die zugehörigen Metadaten verwendet. Diese 32 KB werden gemäß den Gebühren von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive berechnet.

Wenn Sie viele kleinere Objekte speichern, sollten Sie daher keine Lebenszyklusübergaben verwenden. Stattdessen sollten Sie viele kleinere Objekte zu einer kleineren Anzahl von großen Objekten zusammenzufassen, bevor Sie sie in Amazon S3 speichern, um so die Overheadgebühren zu reduzieren. Weitere Informationen zu den Kostenüberlegungen finden Sie unter [Übergang in die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive \(Objektarchivierung\)](#).

- Mindestspeichergebühren

Für einige S3-Speicherklassen gelten Mindestanforderungen an die Speicherdauer. Für Objekte, die vor Ablauf der Mindestdauer gelöscht, überschrieben oder aus diesen Klassen übergeben werden, wird eine anteilige Gebühr für die vorzeitige Übergabe oder das Löschen berechnet. Diese Mindestanforderungen an die Speicherdauer lauten wie folgt:

- S3 Standard-IA und S3 One Zone-IA – 30 Tage
- S3 Glacier Flexible Retrieval und S3 Glacier Instant Retrieval – 90 Tage
- S3 Glacier Deep Archive – 180 Tage

Weitere Informationen zu diesen Anforderungen finden Sie im Abschnitt Beschränkungen unter [Übergang von Objekten mit S3-Lebenszyklus](#). Allgemeine S3-Preisinformationen finden Sie unter [Amazon S3 S3-Preise](#) und [AWS Preisrechner](#).

- Kosten für die Lebenszyklusübergabe

Immer wenn ein Objekt durch eine Lebenszyklusregel in eine andere Speicherkategorie übergeben wird, zählt Amazon S3 diese Übergabe als eine Übergabeanforderung. Die Kosten für diese Übergabeanforderungen kommen zu den Kosten dieser Speicherklassen hinzu. Wenn Sie vorhaben, sehr viele Objekte zu übergeben, sollten Sie die Anforderungskosten für Übergaben in niedrigere Stufen berücksichtigen. Weitere Informationen finden Sie unter [Amazon S3 – Preise](#).

Ich habe meine Bucket-Richtlinie aktualisiert, meine S3-Objekte werden jedoch noch immer aufgrund abgelaufener Lebenszyklusregeln gelöscht.

Deny-Anweisungen in einer Bucket-Richtlinie verhindern den Ablauf der in einer Lebenszyklusregel definierten Objekte nicht. Lebenszyklusaktionen (wie Übergaben oder der Ablauf) verwenden die S3-Operation `DeleteObject` nicht. Stattdessen werden S3-Lebenszyklusaktionen mithilfe interner S3-Endpunkte ausgeführt. (Weitere Informationen finden Sie unter [Lebenszyklus und Protokollieren](#).)

Um zu verhindern, dass Ihre Lebenszyklusregel Aktionen ausführt, müssen Sie die Regel bearbeiten, löschen oder [deaktivieren](#).

Kann ich S3-Objekte wiederherstellen, die aufgrund von S3-Lebenszyklusregeln abgelaufen sind?

Die einzige Möglichkeit, aufgrund des S3-Lebenszyklus abgelaufene Objekte wiederherzustellen, besteht in der Versionsverwaltung. Diese muss vor dem Ablauf der Objekte aktiviert sein. Sie können die Ablaufoperationen, die durch Lebenszyklusregeln ausgeführt werden, nicht rückgängig machen. Wenn Objekte aufgrund der geltenden S3-Lebenszyklusregeln dauerhaft gelöscht werden, können Sie diese Objekte nicht wiederherstellen. Informationen zum Aktivieren der Versionsverwaltung für einen Bucket finden Sie unter [the section called “Verwenden der S3-Versioning”](#).

Wenn Sie die Versionsverwaltung auf den Bucket angewendet haben und die nicht aktuellen Versionen der Objekte noch intakt sind, können Sie [frühere Versionen der abgelaufenen Objekte wiederherstellen](#). Weitere Informationen zum Verhalten der Aktionen von S3-Lebenszyklusregeln und zu Versionsverwaltungsstatus finden Sie in der Tabelle Lebenszyklus-Aktionen und der Versioning-Status eines Buckets in [Elemente, die Lebenszyklus-Aktionen beschreiben](#).

Note

Wenn der S3-Bucket geschützt ist durch [AWS Backup](#) oder [S3-Replikation](#), möglicherweise können Sie diese Funktionen auch verwenden, um Ihre abgelaufenen Objekte wiederherzustellen.

Wie kann ich ein Präfix aus meiner Lebenszyklusregel ausschließen?

S3 Lifecycle unterstützt das Ausschließen von Präfixen in Ihren Regeln nicht. Verwenden Sie stattdessen Tags, um alle Objekte zu kennzeichnen, die Sie in die Regel aufnehmen möchten. Weitere Informationen zur Verwendung von Tags in Ihren Lebenszyklusregeln finden Sie unter [the section called “Beispiel 1: Festlegen eines Filters”](#).

Wie kann ich mehrere Präfixe in meine Lebenszyklusregel aufnehmen?

S3 Lifecycle unterstützt nicht, mehrere Präfixe in Ihre Regeln aufzunehmen. Verwenden Sie stattdessen Tags, um alle Objekte zu kennzeichnen, die Sie in die Regel aufnehmen möchten. Weitere Informationen zur Verwendung von Tags in Ihren Lebenszyklusregeln finden Sie unter [the section called “Beispiel 1: Festlegen eines Filters”](#).

Wenn Sie jedoch über ein oder mehrere Präfixe verfügen, die mit denselben Zeichen beginnen, können Sie all diese Präfixe in Ihre Regel aufnehmen, indem Sie im Filter ein Teilpräfix ohne abschließenden Schrägstrich (/) angeben. Nehmen wir zum Beispiel an, Sie haben diese Präfixe:

```
sales1999/  
sales2000/  
sales2001/
```

Um alle drei Präfixe in Ihre Regel aufzunehmen, geben Sie dies `<Prefix>sales</Prefix>` in Ihrer Lebenszyklusregel an.

Fehlerbehebung bei einer Replikation

In diesem Abschnitt finden Sie Tipps zur Problembehandlung in Amazon S3 Replication und Informationen zu Fehlern in S3 Batch Replication.

Themen

- [Tipps zur Fehlerbehebung in S3 Replication](#)
- [Fehler bei der Batchreplikation](#)

Tipps zur Fehlerbehebung in S3 Replication

Wenn Objektreplikate nicht im Ziel-Bucket erscheinen, nachdem Sie die Replikation konfiguriert haben, können Sie mit diesen Tipps zur Fehlerbehebung Probleme identifizieren und beheben.

- Die meisten Objekte werden innerhalb von 15 Minuten repliziert. Die von Amazon S3 für die Replikation eines Objekts benötigte Zeit hängt von verschiedenen Faktoren ab, einschließlich des Quell- und Ziel-Regionspaars sowie der Größe des Objekts. Bei großen Objekten kann die Replikation mehrere Stunden dauern. Sie können [Begrenzung der S3-Replikationszeit \(S3 RTC\)](#) verwenden, um einen Überblick über die Replikationszeiten zu erhalten.

Wenn das replizierte Objekt groß ist, warten Sie eine Weile, bevor Sie überprüfen, ob es im Ziel erscheint. Sie können auch den Replikationsstatus des Quellobjekts überprüfen. Wenn der Replikationsstatus des Objekts PENDING lautet, hat Amazon S3 die Replikation noch nicht abgeschlossen. Wenn der Replikationsstatus des Objekts FAILED lautet, überprüfen Sie die Replikationskonfiguration des Quell-Buckets. Wenn Sie Informationen zu Fehlern während der Replikation erhalten möchten, können Sie einrichten, dass Sie über Fehlerereignisse bei der Replikation mit Amazon-S3-Ereignisbenachrichtigungen benachrichtigt werden. Weitere

Informationen finden Sie unter [Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse](#).

- Sie können die API-Operation `HeadObject` aufrufen, um den Replikationsstatus eines Objekts zu überprüfen. Die API-Operation `HeadObject` gibt den Replikationsstatus `PENDING`, `COMPLETED` oder `FAILED` eines Objekts zurück. Als Antwort auf den API-Aufruf `HeadObject` wird der Replikationsstatus im Element `x-amz-replication-status` zurückgegeben.

Note

Zum Ausführen von `HeadObject` müssen Sie über Lesezugriff auf das angeforderte Objekt verfügen. Eine `HEAD`-Anforderung bietet die gleichen Optionen wie eine `GET`-Anforderung, ohne eine `GET`-Operation auszuführen. Um beispielsweise eine `HeadObject`-Anforderung mit der AWS Command Line Interface (AWS CLI) auszuführen, können Sie den folgenden Befehl ausführen. Ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api head-object --bucket my-bucket --key index.html
```

- Nachdem `HeadObject` die Objekte mit dem Replikationsstatus `FAILED` zurückgegeben hat, können Sie S3 Batch Replication verwenden, um diese fehlgeschlagenen Objekte zu replizieren. Alternativ können Sie die fehlgeschlagenen Objekte erneut in den Quell-Bucket hochladen, wodurch die Replikation für die neuen Objekte initiiert wird.
- Überprüfen Sie in der Replikations-Konfiguration des Quell-Buckets Folgendes:
 - ob der Amazon-Ressourcennamen (ARN) des Ziel-Buckets korrekt ist.
 - ob das Schlüsselnamenpräfix korrekt ist. Wenn Sie beispielsweise die Konfiguration so einrichten, dass nur Objekte mit dem Präfix `Tax` repliziert werden, werden nur Objekte mit Schlüsselnamen wie beispielsweise `Tax/document1` oder `Tax/document2` repliziert. Ein Objekt mit dem Schlüsselnamen `document3` wird nicht repliziert.
 - Der Status der Replikationsregel lautet `Enabled`.
- Stellen Sie sicher, dass die Versionsverwaltung in der Replikationskonfiguration der Buckets nicht ausgesetzt wurde. Sowohl für die Quell- als auch für die Ziel-Buckets muss die Versionsverwaltung aktiviert sein.
- Wenn eine Replikationsregel auf Objekteigentümerschaft in Eigentümer des Ziel-Buckets ändern gesetzt ist, muss die für die Replikation verwendete AWS Identity and Access Management (IAM)-Rolle über die Berechtigung `s3:ObjectOwnerOverrideToBucketOwner` verfügen.

Diese Berechtigung wird für die Ressource (in diesem Fall den Ziel-Bucket) erteilt. Die folgende Resource-Anweisung zeigt beispielsweise, wie diese Berechtigung für den Ziel-Bucket erteilt wird:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:ObjectOwnerOverrideToBucketOwner"
  ],
  "Resource": "arn:aws:s3:::DestinationBucket/*"
}
```

- Wenn sich der Ziel-Bucket im Besitz eines anderen Kontos befindet, muss der Eigentümer des Ziel-Buckets die Berechtigung `s3:ObjectOwnerOverrideToBucketOwner` über die Ziel-Bucket-Richtlinie auch dem Eigentümer des Quell-Buckets erteilen. Wenn Sie die folgende Bucket-Beispielrichtlinie verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen:

```
{
  "Version": "2012-10-17",
  "Id": "Policy1644945280205",
  "Statement": [
    {
      "Sid": "Stmt1644945277847",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateTags",
        "s3:ObjectOwnerOverrideToBucketOwner"
      ],
      "Resource": "arn:aws:s3:::DestinationBucket/*"
    }
  ]
}
```

Note

Wenn die Einstellungen des Ziel-Buckets zur Objekteigentümerschaft Bucket-Eigentümer erzwungen beinhalten, müssen Sie die Einstellung in der Replikationsregel nicht in

Objekteigentümerschaft in Eigentümer des Ziel-Buckets ändern ändern. Die Änderung der Objekteigentümerschaft erfolgt standardmäßig. Weitere Informationen zum Ändern der Replikateigentümerschaft finden Sie unter [Ändern des Replikat-Eigentümers](#).

- Wenn Sie die Replikationskonfiguration in einem kontoübergreifenden Szenario festlegen, in dem die Quell- und Ziel-Buckets verschiedenen gehören AWS-Konten, können die Ziel-Buckets nicht als Buckets mit Zahlung durch den Anforderer konfiguriert werden. Weitere Informationen finden Sie unter [Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung](#).
- Wenn die Quellobjekte eines Buckets mit einem AWS Key Management Service (AWS KMS)-Schlüssel verschlüsselt sind, muss die Replikationsregel so konfiguriert werden, dass sie mit AWS KMS verschlüsselte Objekte umfasst. Wählen Sie Mit AWS KMS verschlüsselte Objekte replizieren unter den Einstellungen zu Verschlüsselung in der Amazon-S3-Konsole aus. Wählen Sie dann einen AWS KMS-Schlüssel zum Verschlüsseln der Zielobjekte aus.

Note

Wenn sich der Ziel-Bucket in einem anderen Konto befindet, geben Sie einen vom Kunden verwalteten AWS KMS-Schlüssel an, der dem Zielkonto gehört. Verwenden Sie nicht den von Amazon S3 verwalteten Standardschlüssel (aws/s3). Bei Verwendung des Standardschlüssels werden die Objekte mit dem von Amazon S3 verwalteten Schlüssel verschlüsselt, der dem Quellkonto gehört. Dadurch wird verhindert, dass das Objekt für ein anderes Konto freigegeben wird. Infolgedessen kann das Zielkonto nicht auf die Objekte im Ziel-Bucket zugreifen.

Um zum Verschlüsseln der Zielobjekte einen AWS KMS-Schlüssel zu verwenden, der zum Zielkonto gehört, muss das Zielkonto der Replikationsrolle in der KMS-Schlüsselrichtlinie die Berechtigungen `kms:GenerateDataKey` und `kms:Encrypt` erteilen. Wenn Sie die folgende Beispielanweisung in Ihrer KMS-Schlüsselrichtlinie verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen:

```
{
  "Sid": "AllowS3ReplicationSourceRoleToUseTheKey",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
  },
}
```

```

    "Action": ["kms:GenerateDataKey", "kms:Encrypt"],
    "Resource": "*"
  }

```

Wenn Sie ein Sternchen (*) für die Resource-Anweisung in der AWS KMS-Schlüsselrichtlinie verwenden, erteilt die Richtlinie nur der Replikationsrolle die Berechtigung, den KMS-Schlüssel zu verwenden. Die Richtlinie erlaubt der Replikationsrolle nicht, ihre Berechtigungen zu erweitern.

Standardmäßig gewährt die KMS-Schlüsselrichtlinie dem Root-Benutzer volle Berechtigungen für den Schlüssel. Diese Berechtigungen können an andere Benutzer in demselben Konto delegiert werden. Sofern die Quell-KMS-Schlüsselrichtlinie keine Deny-Anweisungen enthält, reicht es aus, eine IAM-Richtlinie zu verwenden, um der Replikationsrolle Berechtigungen für den Quell-KMS-Schlüssel zu erteilen.

Note

KMS-Schlüsselrichtlinien, die den Zugriff auf bestimmte CIDR-Bereiche, VPC-Endpunkte oder S3-Zugriffspunkte einschränken, können dazu führen, dass die Replikation fehlschlägt.

Wenn je nach Verschlüsselungskontext entweder der Quell- oder der Ziel-KMS-Schlüssel Berechtigungen erteilt, vergewissern Sie sich, dass Amazon-S3-Bucket-Schlüssel für die Buckets aktiviert sind. Wenn die S3-Bucket-Schlüssel für die Buckets aktiviert sind, muss der Verschlüsselungskontext die Ressource auf Bucket-Ebene sein, wie im Folgenden veranschaulicht:

```

"kms:EncryptionContext:arn:aws:arn": [
  "arn:aws:s3:::SOURCE_BUCKET_NAME"
]
"kms:EncryptionContext:arn:aws:arn": [
  "arn:aws:s3:::DESTINATION_BUCKET_NAME"
]

```

Zusätzlich zu den durch die KMS-Schlüsselrichtlinie gewährten Berechtigungen muss das Quellkonto der IAM-Richtlinie der Replikationsrolle die folgenden Mindestberechtigungen hinzufügen:

```

{
  "Effect": "Allow",

```

```

    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": [
      "SourceKmsKeyArn"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:GenerateDataKey",
      "kms:Encrypt"
    ],
    "Resource": [
      "DestinationKmsKeyArn"
    ]
  }
}

```

Weitere Informationen zur Replikation von mit AWS KMS verschlüsselten Objekten finden Sie unter [Replizieren verschlüsselter Objekte](#).

- Wenn sich der Ziel-Bucket im Besitz eines anderen AWS-Konto befindet, stellen Sie sicher, dass der Bucket-Eigentümer eine Bucket-Richtlinie für den Ziel-Bucket eingerichtet hat, die dem Eigentümer des Quell-Buckets die Replikation von Objekten gestattet. Ein Beispiel finden Sie unter [Konfigurieren der Replikation, wenn sich Quell- und Ziel-Buckets im Eigentum verschiedener Konten befinden](#).
- Wenn Ihre Objekte auch nach der Bestätigung der Berechtigungen nicht repliziert werden, suchen Sie an den folgenden Stellen nach expliziten Deny-Anweisungen:
 - Deny-Anweisungen in den Richtlinien des Quell- oder Ziel-Buckets. Die Replikation schlägt fehl, wenn die Bucket-Richtlinie den Zugriff auf die Replikationsrolle für eine der folgenden Aktionen verweigert:

Quell-Bucket

```

"s3:GetReplicationConfiguration",
"s3:ListBucket",
"s3:GetObjectVersionForReplication",
"s3:GetObjectVersionAcl",
"s3:GetObjectVersionTagging"

```

Ziel-Buckets:

```
"s3:ReplicateObject",  
"s3:ReplicateDelete",  
"s3:ReplicateTags"
```

- Deny-Anweisungen oder Berechtigungsgrenzen, die der IAM-Rolle zugeordnet sind, können dazu führen, dass die Replikation fehlschlägt.
- Deny-Anweisungen in den Service-Kontrollrichtlinien von AWS Organizations, die entweder dem Quell- oder dem Zielkonto zugeordnet sind, können dazu führen, dass die Replikation fehlschlägt.
- Wenn ein Objektrepikat nicht im Ziel-Bucket angezeigt wird, könnten folgende Probleme die Replikation verhindert haben:
 - Amazon S3 repliziert keine Objekte in einem Quell-Bucket, der ein Replikat ist, das mit einer anderen Replikations-Konfiguration erstellt wurde. Wenn Sie beispielsweise die Replikationskonfiguration von Bucket A zu Bucket B zu Bucket C festlegen, repliziert Amazon S3 keine Objektrepikate von Bucket B zu Bucket C.
 - Ein Quell-Bucket-Eigentümer kann anderen AWS-Konten Berechtigungen für das Hochladen von Objekten erteilen. Standardmäßig besitzt der Quell-Bucket-Eigentümer keine Berechtigungen für die Objekte, die von anderen Konten erstellt wurden. Die Replikations-Konfiguration repliziert nur die Objekte, für die der Quell-Bucket-Eigentümer über Zugriffsberechtigungen verfügt. Der Bucket-Eigentümer kann anderen AWS-Konten Berechtigungen zum bedingten Erstellen von Objekten erteilen. Dabei sind explizite Zugriffsberechtigungen für diese Objekte erforderlich. Eine Beispielrichtlinie finden Sie unter [Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt](#).
- Angenommen, Sie fügen einer Replikationskonfiguration eine Regel hinzu, um eine Teilmenge von Objekten mit einem spezifischen Tag zu replizieren. In diesem Fall müssen Sie den spezifischen Tag-Schlüssel und -Wert zum Zeitpunkt der Objekterstellung zuweisen, damit Amazon S3 das Objekt replizieren kann. Wenn Sie zuerst ein Objekt erstellen und dann dem vorhandenen Objekt das Tag hinzufügen, repliziert Amazon S3 das Objekt nicht.

- Verwenden Sie Amazon-S3-Ereignisbenachrichtigungen, um sich informieren zu lassen, wenn Objekte nicht in ihre Ziel-AWS-Region repliziert wurden. Amazon-S3-Ereignisse sind über Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) oder AWS Lambda verfügbar. Weitere Informationen finden Sie unter [Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse](#).

Sie können Amazon-S3-Ereignisbenachrichtigungen auch verwenden, um die Ursachen für fehlgeschlagene Replikationen anzuzeigen. Eine Liste der Ursachen für fehlgeschlagene Replikationen finden Sie unter [Gründe für das Fehlschlagen der Replikation in Amazon S3](#).

Fehler bei der Batchreplikation

Um Fehler bei Objekten zu beheben, die nicht in den Ziel-Bucket repliziert werden, überprüfen Sie die verschiedenen Arten von Berechtigungen für den Bucket, die Replikationsrolle und die IAM-Rolle, die zum Erstellen des Batchreplikationsauftrags verwendet werden. Überprüfen Sie außerdem die Einstellungen für den öffentlichen Zugriff und die Einstellungen für die Bucket-Eigentümerschaft.

Während der Batchreplikation können die folgenden Fehler auftreten:

- Die Batchoperation ist aus folgendem Grund fehlgeschlagen: Der Auftragsbericht konnte nicht in Ihren Berichts-Bucket geschrieben werden.

Dieser Fehler tritt auf, wenn die für den Batch-Operations-Auftrag verwendete IAM-Rolle den Abschlussbericht nicht an dem Speicherort speichern kann, der während der Auftragserstellung angegeben wurde. Überprüfen Sie zum Beheben dieses Fehlers, ob die IAM-Rolle über PutObject-Berechtigungen für den Bucket verfügt, in dem der Batch-Operations-Abschlussbericht gespeichert werden soll. Eine bewährte Methode besteht darin, den Bericht in einem anderen Bucket als dem Quell-Bucket bereitzustellen.

- Die Batchoperation wurde mit Fehlern abgeschlossen und die Gesamtzahl der Fehlschläge ist nicht 0.

Dieser Fehler tritt auf, wenn bei dem ausgeführten Batchreplikationsauftrag unzureichende Objektberechtigungen vorhanden sind. Wenn Sie eine Replikationsregel für Ihren Batchreplikationsauftrag verwenden, stellen Sie sicher, dass die für die Replikation verwendete IAM-Rolle über die richtigen Berechtigungen für den Zugriff auf Objekte aus dem Quell- oder Ziel-Bucket verfügt. Sie können auch den [Abschlussbericht zur Batchreplikation](#) anzeigen, um den spezifischen [Grund für das Fehlschlagen der Replikation in Amazon S3](#) zu überprüfen.

- Der Batchauftrag wurde erfolgreich ausgeführt, die Anzahl der im Ziel-Bucket erwarteten Objekte stimmt nicht überein.

Dieser Fehler tritt auf, wenn die im Manifest des Batchreplikationsauftrags aufgeführten Objekte nicht mit den Filtern übereinstimmen, die Sie während der Auftragserstellung ausgewählt haben. Möglicherweise erhalten Sie diese Meldung auch, wenn die Objekte in Ihrem Quell-Bucket keinen Replikationsregeln entsprechen und nicht im generierten Manifest enthalten sind.

Behebung von Fehlern bei der Server-Zugriffsprotokollierung

Die folgenden Themen können Ihnen bei der Behebung von Problemen helfen, die beim Einrichten der Protokollierung mit Amazon S3 auftreten können.

Themen

- [Häufige Fehlermeldungen beim Einrichten der Protokollierung](#)
- [Behebung von Bereitstellungsfehlern](#)

Häufige Fehlermeldungen beim Einrichten der Protokollierung

Die folgenden häufigen Fehlermeldungen können angezeigt werden, wenn Sie die Protokollierung über die AWS Command Line Interface (AWS CLI) und AWS-SDKs aktivieren:

Fehler: S3-standortübergreifende Protokollierung nicht zulässig

Wenn sich der Ziel-Bucket in einer anderen Region als der Quell-Bucket befindet, tritt der Fehler S3-standortübergreifende Protokollierung nicht zulässig auf. Um diesen Fehler zu beheben, stellen Sie sicher, dass sich der Ziel-Bucket, der für den Empfang der Zugriffsprotokolle konfiguriert ist, in derselben AWS-Region und demselben AWS-Konto wie der Quell-Bucket befindet.

Fehler: Die Eigentümer des zu protokollierenden Buckets und des Ziel-Buckets müssen identisch sein

Wenn Sie die Server-Zugriffsprotokollierung aktivieren, tritt dieser Fehler auf, wenn der angegebene Ziel-Bucket zu einem anderen Konto gehört. Um diesen Fehler zu beheben, stellen Sie sicher, dass sich der Ziel-Bucket in demselben AWS-Konto wie der Quell-Bucket befindet.

Note

Wir empfehlen, einen anderen Ziel-Bucket als den Quell-Bucket auszuwählen. Wenn Quell-Bucket und Ziel-Bucket identisch sind, werden für die Protokolle, die in den Bucket

geschrieben werden, zusätzliche Protokolle erstellt, wodurch die Speicherkosten steigen können. Diese zusätzlichen Protokolle zu den Protokollen können es zudem schwierig machen, die gesuchten Protokolle zu finden. Zur einfacheren Protokollverwaltung empfehlen wir, Zugriffsprotokolle in einem anderen Bucket zu speichern. Weitere Informationen finden Sie unter [the section called “Wie aktiviere ich die Protokollzustellung?”](#).

Fehler: Der Ziel-Bucket für die Protokollierung ist nicht vorhanden

Der Ziel-Bucket muss vorhanden sein, bevor die Konfiguration festgelegt wird. Dieser Fehler weist darauf hin, dass der Ziel-Bucket nicht vorhanden ist oder nicht gefunden werden kann. Vergewissern Sie sich, dass der Bucket-Name richtig geschrieben ist, und wiederholen Sie den Vorgang.

Fehler: Ziel-Erteilungen sind für Buckets mit „Bucket-Eigentümer erzwungen“ nicht zulässig

Dieser Fehler weist darauf hin, dass der Ziel-Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für die S3-Objekt-Eigentümerschaft verwendet. Die Einstellung „Vom Bucket-Eigentümer erzwungen“ unterstützt keine Ziel-Erteilungen. Weitere Informationen finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Behebung von Bereitstellungsfehlern

Halten Sie sich an die folgenden bewährten Methoden, um Probleme mit der Server-Zugriffsprotokollierung zu vermeiden:

- Die S3-Protokollbereitstellungsgruppe hat Schreibzugriff auf den Ziel-Bucket – Die S3-Protokollbereitstellungsgruppe liefert Serverzugriffsprotokolle an den Ziel-Bucket. Es kann eine Bucket-Richtlinie oder eine Bucket-Zugriffssteuerungsliste (ACL) verwendet werden, um Schreibzugriff auf den Ziel-Bucket zu gewähren. Wir empfehlen jedoch, eine Bucket-Richtlinie anstelle einer ACL zu verwenden. Weitere Informationen zum Gewähren von Schreibzugriff für Ihren Ziel-Bucket finden Sie unter [Berechtigungen für Protokollbereitstellung](#).

Note

Wenn der Ziel-Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für die Objekt-Eigentümerschaft verwendet, beachten Sie Folgendes:

- ACLs sind deaktiviert und wirken sich nicht mehr auf die Berechtigungen aus. Somit können Sie Ihre Bucket-ACL nicht aktualisieren, um Zugriff auf die S3-Protokollbereitstellungsgruppe zu gewähren. Sie müssen vielmehr die Bucket-

Richtlinie für den Ziel-Bucket aktualisieren, um Zugriff auf den Prinzipal des Protokollierungsservices zu gewähren.

- Sie können keine Ziel-Erteilungen in Ihre PutBucketLogging-Konfiguration einschließen.

- Die Bucket-Richtlinie für den Ziel-Bucket lässt den Zugriff auf die Protokolle zu – Überprüfen Sie die Bucket-Richtlinie des Ziel-Buckets. Suchen Sie in der Bucket-Richtlinie nach allen Anweisungen, die "Effect": "Deny" enthalten. Stellen Sie anschließend sicher, dass die Deny-Anweisung das Schreiben von Zugriffsprotokollen in den Bucket nicht verhindert.
- S3 Object Lock ist auf dem Ziel-Bucket nicht aktiviert – Prüfen Sie, ob Object Lock für den Ziel-Bucket aktiviert ist. Die Objektsperre blockiert die Bereitstellung von Server-Zugriffsprotokollen. Sie müssen einen Ziel-Bucket auswählen, für den Object Lock nicht aktiviert ist.
- Von Amazon S3 verwaltete Schlüssel (SSE-S3) sind ausgewählt, wenn die Standardverschlüsselung auf dem Ziel-Bucket aktiviert ist – Sie können die Bucket-Standardverschlüsselung nur für den Ziel-Bucket verwenden, wenn Sie die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) verwenden. Die standardmäßige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS)-Schlüsseln (SSE-KMS) wird für Ziel-Buckets für die Server-Zugriffsprotokollierung nicht unterstützt. Weitere Informationen zum Aktivieren der Standardverschlüsselung finden Sie unter [Konfigurieren der Standardverschlüsselung](#).
- Für den Ziel-Bucket ist „Zahlung durch den Anforderer“ nicht aktiviert – Die Verwendung eines Buckets mit „Zahlung durch den Anforderer“ als Ziel-Bucket für die Server-Zugriffsprotokollierung wird nicht unterstützt. Um die Bereitstellung von Serverzugriffsprotokollen zu ermöglichen, deaktivieren Sie die Option „Zahlung durch den Anforderer“ im Ziel-Bucket.
- Überprüfen Sie Ihre AWS Organizations-Service-Kontrollrichtlinie – Überprüfen Sie bei Verwendung von AWS Organizations die Service-Kontrollrichtlinien, um sicherzustellen, dass der Amazon-S3-Zugriff zulässig ist. Service-Kontrollrichtlinien geben die maximalen Berechtigungen für die betroffenen Konten an. Suchen Sie in der Service-Kontrollrichtlinie nach Anweisungen, die "Effect": "Deny" enthalten, und stellen Sie sicher, dass Deny-Anweisungen das Schreiben von Zugriffsprotokollen in den Bucket nicht verhindern. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien \(SCPs\)](#) im AWS Organizations-Benutzerhandbuch.
- Eine gewisse Zeit warten, bis die letzten Änderungen an der Protokollierungskonfiguration wirksam werden – Die erstmalige Aktivierung der Server-Zugriffsprotokollierung oder die Änderung des Ziel-Buckets für Protokolle wird erst nach einer gewissen Zeit vollständig wirksam. Es kann über eine Stunde dauern, bis alle Anforderungen ordnungsgemäß protokolliert und übermittelt wurden.

Um nach Fehlern bei der Protokollzustellung zu suchen, aktivieren Sie Anforderungsmetriken in Amazon CloudWatch. Wenn die Protokolle nicht innerhalb weniger Stunden bereitgestellt werden, suchen Sie nach der Metrik `4xxErrors`, die auf Protokollbereitstellungsfehler hinweisen kann. Weitere Informationen zur Aktivierung von Anforderungsmetriken finden Sie unter [the section called “Erstellen einer Metrik-Konfiguration für alle Objekte”](#).

Fehlerbehebung für die Versionsverwaltung

Die folgenden Themen können bei der Behebung von allgemeinen Problemen mit der Amazon-S3-Versionsverwaltung hilfreich sein.

Themen

- [Ich möchte Objekte wiederherstellen, die in einem Bucket mit aktivierter Versionsverwaltung versehentlich gelöscht wurden.](#)
- [Ich möchte versionierte Objekte dauerhaft löschen](#)
- [Nach dem Aktivieren der Bucket-Versionsverwaltung stelle ich Leistungseinbußen fest](#)

Ich möchte Objekte wiederherstellen, die in einem Bucket mit aktivierter Versionsverwaltung versehentlich gelöscht wurden.

Wenn Objektversionen aus S3-Buckets gelöscht werden, hat Amazon S3 im Allgemeinen keine Möglichkeit, diese wiederherzustellen. Wenn Sie jedoch die S3-Versionsverwaltung in Ihrem S3-Bucket aktiviert haben, kann ein Objekt mit einer DELETE-Anforderung, in der keine Versions-ID angegeben ist, nicht dauerhaft gelöscht werden. Stattdessen wird eine Löschmarkierung als Platzhalter hinzugefügt. Diese Löschmarkierung wird zur aktuellen Version des Objekts.

Gehen Sie wie folgt vor, um zu überprüfen, ob Ihre gelöschten Objekte dauerhaft oder vorübergehend (mit einer Löschmarkierung an ihrer Stelle) gelöscht wurden:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.

4. Aktivieren Sie in der Liste Objekte den Schalter Versionen anzeigen rechts neben der Suchleiste und suchen Sie dann in der Suchleiste nach dem gelöschten Objekt. Dieser Schalter ist nur verfügbar, wenn die Versionsverwaltung zuvor für den Bucket aktiviert wurde.

Sie können auch das [S3 Inventory verwenden, um nach gelöschten Objekten zu suchen](#).

5. Wenn Sie das Objekt nicht finden können, nachdem Sie Versionen anzeigen eingeschaltet oder einen Bestandsbericht erstellt haben, und auch keine [Löschmarkierung](#) für das Objekt zu finden ist, ist der Löschvorgang dauerhaft und das Objekt kann nicht wiederhergestellt werden.

Sie können den Status eines gelöschten Objekts auch überprüfen, indem Sie den HeadObject API Vorgang von AWS Command Line Interface (AWS CLI) aus verwenden. Verwenden Sie hierfür den folgenden Befehl `head-object` und ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html
```

Wenn Sie den Befehl `head-object` für ein versioniertes Objekt ausführen, dessen aktuelle Version eine Löschmarkierung ist, erhalten Sie den Fehler 404 Nicht gefunden. Beispielsweise:

Beim Aufrufen der HeadObject Operation ist ein Fehler aufgetreten (404): Not Found

Wenn Sie den Befehl `head-object` für ein versioniertes Objekt ausführen und die Versions-ID des Objekts angeben, ruft Amazon S3 die Metadaten des Objekts ab und bestätigt damit, dass das Objekt noch vorhanden ist und nicht dauerhaft gelöscht wurde.

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html --  
version-id versionID
```

```
{  
  "AcceptRanges": "bytes",  
  "ContentType": "text/html",  
  "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",  
  "ContentLength": 77,  
  "VersionId": "Zg5HyL7m.eZU9iM7AV1JkrqAiE.0UG4q",  
  "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",  
  "Metadata": {}  
}
```

Wenn das Objekt gefunden wird und die neueste Version eine Löschmarkierung ist, ist die vorherige Version des Objekts immer noch vorhanden. Da die Löschmarkierung die aktuelle Version des Objekts darstellt, können Sie das Objekt wiederherstellen, indem Sie die Löschmarkierung löschen.

Nachdem Sie die Löschmarkierung dauerhaft entfernt haben, wird die zweitneueste Version des Objekts zur aktuellen Version des Objekts, sodass Ihr Objekt wieder verfügbar ist. Eine visuelle Darstellung der Wiederherstellung von Objekten finden Sie unter [Entfernen von Löschmarkierungen](#).

Um eine bestimmte Version eines Objekts entfernen zu können, müssen Sie der Bucket-Eigentümer sein. Um eine Löschmarkierung dauerhaft zu löschen, müssen Sie Ihre Versions-ID in einer DeleteObject-Anforderung angeben. Verwenden Sie zum Löschen der Löschmarkierung den folgenden Befehl und ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen:

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key index.html --  
version-id versionID
```

Weitere Informationen über den Befehl `delete-object` finden Sie unter [delete-object](#) in der AWS CLI -Befehlsreferenz. Weitere Informationen zum dauerhaften Löschen von Löschmarkierungen finden Sie unter [Verwalten von Löschmarkierungen](#).

Ich möchte versionierte Objekte dauerhaft löschen

In einem Bucket mit aktivierter Versionsverwaltung kann ein Objekt mit einer DELETE-Anforderung ohne Versions-ID nicht dauerhaft gelöscht werden. Bei einer solchen Anforderung wird vielmehr eine Löschmarkierung eingefügt.

Um versionierte Objekte dauerhaft zu löschen, können Sie eine der folgenden Methoden auswählen:

- Erstellen Sie eine S3-Lebenszyklusregel, um nicht aktuelle Versionen dauerhaft zu löschen. Um nicht aktuelle Versionen dauerhaft zu löschen, wählen Sie Nicht aktuelle Versionen von Objekten dauerhaft löschen aus und geben Sie anschließend eine Zahl unter Tage, nachdem Objekte veraltet sind ein. Sie können die Anzahl der beizubehaltenden neueren Versionen optional angeben, indem Sie einen Wert unter Number of newer versions to retain (Anzahl der beizubehaltenden neueren Versionen) eingeben. Weitere Informationen zum Erstellen dieser Regel finden Sie unter [Festlegen einer S3-Lebenszyklus-Konfiguration](#).
- Löschen Sie eine angegebene Version, indem Sie die Versions-ID in die DELETE-Anforderung aufnehmen. Weitere Informationen hierzu finden Sie unter [Dauerhaftes Löschen von versionierten Objekten](#).

- Erstellen Sie eine Lebenszyklusregel, um aktuelle Versionen ablaufen zu lassen. Um aktuelle Versionen von Objekten ablaufen zu lassen, wählen Sie Aktuelle Objektversionen ablaufen lassen aus und fügen Sie dann eine Zahl unter Tage nach Objekterstellung ein. Weitere Informationen zum Erstellen dieser Lebenszyklusregel finden Sie unter [Festlegen einer S3-Lebenszyklus-Konfiguration](#).
- Um alle versionierten Objekte und Löschmarkierungen dauerhaft zu löschen, erstellen Sie zwei Lebenszyklusregeln: eine, um aktuelle Versionen ablaufen zu lassen und nicht aktuelle Versionen von Objekten dauerhaft zu löschen, und die andere, um Löschmarkierungen für abgelaufene Objekte zu löschen.

In einem Bucket mit aktivierter Versionsverwaltung können mit einer DELETE-Anforderung, in der keine Versions-ID angegeben ist, nur Objekte mit einer Versions-ID NULL entfernt werden. Wenn das Objekt hochgeladen wurde, als die Versionsverwaltung aktiviert war, wird mit einer DELETE-Anforderung, in der keine Versions-ID angegeben ist, eine Löschmarkierung für dieses Objekt erstellt.

Note

Bei Buckets mit aktivierter S3-Objektsperre verursacht eine DELETE-Objektanforderung mit einer ID einer geschützten Objektversion den Fehler 403 Zugriff verweigert. Eine DELETE-Objektanforderung ohne Versions-ID fügt eine Löschmarkierung als neueste Version des Objekts mit der Antwort 200 OK hinzu. Objekte, die durch die Objektsperre geschützt sind, können erst dauerhaft gelöscht werden, wenn ihre Aufbewahrungszeiträume und gesetzlichen Aufbewahrungsfristen aufgehoben sind. Weitere Informationen finden Sie unter [the section called “So funktioniert die S3-Objektsperre”](#).

Nach dem Aktivieren der Bucket-Versionsverwaltung stelle ich Leistungseinbußen fest

Bei Buckets mit aktivierter Versionsverwaltung kann es zu Leistungseinbußen kommen, wenn zu viele Löschmarkierungen oder versionierte Objekte vorhanden sind und wenn bewährte Methoden nicht befolgt werden.

Zu viele Löschmarkierungen

Nachdem Sie die Versionsverwaltung für einen Bucket aktiviert haben, wird mit einer DELETE-Anforderung ohne Versions-ID an ein Objekt eine Löschmarkierung mit einer eindeutigen Versions-

ID erstellt. Lebenszyklus-Konfigurationen mit der Regel Aktuelle Objektversionen ablaufen lassen fügen jedem Objekt eine Löschmarkierung mit einer eindeutigen Versions-ID hinzu. Wenn zu viele Löschmarkierungen vorhanden sind, kann dies die Leistung im Bucket beeinträchtigen.

Wenn die Versionsverwaltung für einen Bucket ausgesetzt wird, markiert Amazon S3 die Versions-ID bei neu erstellten Objekten als NULL. Eine Ablaufaktion in einem Bucket mit ausgesetzter Versionsverwaltung bewirkt, dass Amazon S3 eine Löschmarkierung mit der Versions-ID NULL erstellt. In einem Bucket mit ausgesetzter Versionsverwaltung wird für jede Löschanforderung eine NULL-Löschmarkierung erstellt. Diese NULL-Löschmarkierungen werden auch als Löschmarkierungen für abgelaufenes Objekt bezeichnet, wenn alle Objektversionen gelöscht werden und nur eine einzelne Löschmarkierung übrig bleibt. Wenn sich zu viele NULL-Löschmarkierungen ansammeln, kommt es zu Leistungseinbußen im Bucket.

Zu viele versionierte Objekte

Wenn ein Bucket mit aktivierter Versionsverwaltung Objekte mit Millionen von Versionen enthält, kann es vermehrt zu Fehlern 503 Service nicht verfügbar kommen. Wenn Sie feststellen, dass die Anzahl der HTTP 503 Antworten „Service Unavailable“ für PUT oder DELETE Objektanfragen an einen Bucket mit aktivierter Versionierung deutlich gestiegen ist, befinden sich in dem Bucket möglicherweise ein oder mehrere Objekte mit Millionen von Versionen. Wenn Sie Objekte mit Millionen Versionen haben, drosselt Amazon S3 Anforderungen an den Bucket automatisch. Durch die Drosselung von Anforderungen wird Ihr Bucket vor übermäßigem Anforderungsdatenverkehr geschützt, der andere Anforderungen an denselben Bucket verhindern könnte.

Um festzustellen, zu welchen Objekten es Millionen Versionen gibt, verwenden Sie S3 Inventory. S3 Inventory generiert einen Bericht, der eine flache Dateiliste der Objekte in einem Bucket enthält. Weitere Informationen finden Sie unter [Amazon S3 Inventory](#).

Um zu überprüfen, ob der Bucket viele versionierte Objekte enthält, verwenden Sie die S3-Storage-Lens-Metriken, um die Anzahl der Objekte der aktuellen Version, die Anzahl nicht aktueller Versionsobjekte und die Anzahl der Löschmarkierungsobjekte anzuzeigen. Weitere Informationen zu Storage-Lens-Metriken finden Sie unter [Amazon S3-Storage-Lens-Metrik glossar](#).

Das Amazon-S3-Team fordert die Kunden auf, Anwendungen zu überprüfen, die wiederholt dasselbe Objekt überschreiben und damit potenziell Millionen Versionen für dieses Objekt erstellen, um festzustellen, ob die Anwendung wie beabsichtigt funktioniert. Beispielsweise kann eine Anwendung, die eine Woche lang jede Minute dasselbe Objekt überschreibt, über zehntausend Versionen erstellen. Wir empfehlen, für jedes Objekt weniger als einhunderttausend Versionen zu speichern. Wenn Sie einen Anwendungsfall haben, der Millionen von Versionen für ein oder mehrere Objekte

erfordert, wenden Sie sich an das AWS Support Team, um Unterstützung bei der Suche nach einer besseren Lösung zu erhalten.

Bewährte Methoden

Um Leistungseinbußen im Zusammenhang mit der Versionsverwaltung zu vermeiden, empfehlen wir Ihnen, den folgenden bewährten Methoden zu folgen:

- Aktivieren Sie eine Lebenszyklusregel, um die vorherigen Versionen von Objekten ablaufen zu lassen. Sie können beispielsweise eine Lebenszyklusregel erstellen, nach der nicht aktuelle Versionen ablaufen, wenn das Objekt 30 Tage nicht mehr aktuell ist. Sie können auch mehrere nicht aktuelle Versionen beibehalten, wenn Sie nicht alle löschen möchten. Weitere Informationen finden Sie unter [Festlegen einer S3-Lebenszyklus-Konfiguration](#).
- Aktivieren Sie eine Lebenszyklusregel, um Löschmarkierungen für abgelaufene Objekte zu löschen, denen keine Datenobjekte im Bucket zugeordnet sind. Weitere Informationen finden Sie unter [Löschen abgelaufener Löschmarkierungen für Objekte](#).

Weitere bewährte Methoden zur Amazon-S3-Leistungsoptimierung finden Sie unter [Bewährte Methoden für Designmuster](#).

Ich erhalte eine Amazon S3 S3-Anfrage IDs für AWS Support

Wann immer Sie Kontakt aufnehmen AWS Support , weil Sie auf Fehler oder unerwartetes Verhalten in Amazon S3 gestoßen sind, müssen Sie die Anfrage angeben, die mit der fehlgeschlagenen Aktion IDs verknüpft ist. AWS Support verwendet diese AnfrageIDs, um Ihnen bei der Lösung der Probleme zu helfen, die bei Ihnen aufgetreten sind.

Anfragen IDs kommen paarweise, werden in jeder Antwort zurückgegeben, die Amazon S3 verarbeitet (auch in den fehlerhaften), und sie können über ausführliche Protokolle abgerufen werden. Es gibt eine Reihe gängiger Methoden, um Ihre Anfrage zu erhaltenIDs, darunter S3-Zugriffsprotokolle und AWS CloudTrail Ereignisse oder Datenereignisse.

Nachdem Sie diese Protokolle wiederhergestellt haben, kopieren Sie diese beiden Werte und behalten Sie sie bei, da Sie sie benötigen, wenn Sie Kontakt aufnehmen AWS Support. Informationen zur Kontaktaufnahme finden Sie AWS Support unter [Kontakt AWS](#) oder in der [AWS Support Dokumentation](#).

Wird verwendetHTTP, um eine Anfrage zu erhalten IDs

Sie können Ihre Anfrage abrufenIDs, `x-amz-request-id` `x-amz-id-2` indem Sie die einzelnen Teile einer HTTP Anfrage protokollieren, bevor sie die Zielanwendung erreicht. Es gibt eine Vielzahl von Tools von Drittanbietern, mit denen ausführliche Protokolle für HTTP Anfragen wiederhergestellt werden können. Wählen Sie einen aus, dem Sie vertrauen, und führen Sie dann das Tool aus, um den Port abzuhören, über den Ihr Amazon S3 S3-Verkehr übertragen wird, während Sie eine weitere Amazon S3 HTTP S3-Anfrage senden.

Bei HTTP Anfragen sieht das IDs Anforderungspaar wie folgt aus:

```
x-amz-request-id: 79104EXAMPLEB723
x-amz-id-2: IOwQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km
```

Note

HTTPSAnfragen sind verschlüsselt und in den meisten Paketerfassungen versteckt.

Verwenden Sie einen Webbrowser, um eine Anfrage zu erhalten IDs

Die meisten Webbrowser beinhalten Entwickler-Tools, mit denen Sie Anforderungs-Header anzeigen können.

Bei webbrowserbasierten Anfragen, die einen Fehler zurückgeben, sieht das IDs Anforderungspaar wie in den folgenden Beispielen aus.

```
<Error><Code>AccessDenied</Code><Message>Access Denied</Message>
<RequestId>79104EXAMPLEB723</RequestId><HostId>IOwQ4fDEXAMPLEQM
+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km</HostId></Error>
```

Um das Anforderungs-ID-Paar aus erfolgreichen Anfragen zu erhalten, verwenden Sie die Entwicklertools Ihres Browsers, um sich die HTTP Antwort-Header anzusehen. Informationen zu Entwicklertools für bestimmte Browser finden Sie unter [Amazon S3 S3-Fehlerbehebung](#) — So stellen Sie Ihre S3-Anfrage wieder her IDs in [AWS re:Post](#).

Verwenden Sie die AWS SDKs, um eine Anfrage zu erhalten IDs

Die folgenden Abschnitte enthalten Informationen zur Konfiguration der Protokollierung mithilfe von AWS SDK. Auch wenn Sie die Verbose-Protokollierung für jede Anforderung und jede Antwort aktivieren können, sollten Sie die Protokollierung nicht in Produktionssystemen aktivieren, da große Anforderungen und Antworten eine Anwendung erheblich verlangsamen können.

Bei AWS SDK Anfragen sieht das IDs Anforderungspaar wie in den folgenden Beispielen aus.

```
Status Code: 403, AWS Service: Amazon S3, AWS Request ID: 79104EXAMPLEB723
AWS Error Code: AccessDenied AWS Error Message: Access Denied
S3 Extended Request ID: I0WQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK
+Jd1vEXAMPLEa3Km
```

Verwenden Sie SDK for Go, um eine Anfrage zu erhalten IDs

Sie können die Protokollierung mithilfe von SDK for Go konfigurieren. Weitere Informationen finden Sie unter [Antwortmetadaten](#) im SDKfor Go V2 Developer Guide.

Verwenden Sie das SDK forPHP, um eine Anfrage zu erhalten IDs

Sie können die Protokollierung konfigurieren, indem Sie PHP Weitere Informationen finden Sie unter [Wie kann ich sehen, welche Daten übertragen wurden?](#) im Entwicklerhandbuch zum AWS SDK for PHP .

Verwenden von SDK for Java, um eine Anfrage zu erhalten IDs

Sie können die Protokollierung für spezifische Anforderungen oder Antworten aktivieren, um nur die relevanten Header zu erfassen und zurückzugeben. Dazu importieren Sie die Klasse `com.amazonaws.services.s3.S3ResponseMetadata`. Anschließend können Sie die Anforderung in einer Variablen speichern, bevor Sie die eigentliche Anforderung ausführen. Rufen Sie `getCachedResponseMetadata(AmazonWebServiceRequest request).getRequestID()` auf, um die protokollierte Anforderung oder Antwort zu erhalten.

Example

```
PutObjectRequest req = new PutObjectRequest(bucketName, key, createSampleFile());
s3.putObject(req);
S3ResponseMetadata md = s3.getCachedResponseMetadata(req);
System.out.println("Host ID: " + md.getHostId() + " RequestID: " + md.getRequestId());
```


Alternativ können Sie eine Verbose-Protokollierung jeder Java-Anfrage und -Antwort verwenden. Weitere Informationen finden Sie unter [Verbose-Protokollierung des Netzwerkverkehrs](#) im Entwicklerhandbuch zu AWS SDK for Java .

Verwenden von AWS SDK for .NET , um eine Anfrage zu erhalten IDs

Sie können die Protokollierung mit dem konfigurieren, AWS SDK for .NET indem Sie das integrierte System.Diagnostics Protokollierungstool verwenden. Weitere Informationen finden Sie unter [Protokollierung mit dem AWS SDK für .NET](#) AWS Blogbeitrag für Entwickler.

Note

Standardmäßig enthält das zurückgegebene Protokoll nur Fehlerinformationen. Um die Anfrage zu erhalten IDs, muss die Konfigurationsdatei `AWSLogMetrics` (und optional `AWSResponseLogging`) hinzugefügt worden sein.

Verwenden von SDK for Python (Boto3), um eine Anfrage zu erhalten IDs

Mit dem AWS SDK for Python (Boto3) können Sie bestimmte Antworten protokollieren. Sie können diese Funktion verwenden, um nur die relevanten Header zu erfassen. Der folgende Code zeigt, wie Teile der Antwort in einer Datei protokolliert werden:

```
import logging
import boto3
logging.basicConfig(filename='logfile.txt', level=logging.INFO)
logger = logging.getLogger(__name__)
s3 = boto3.resource('s3')
response = s3.Bucket(bucket_name).Object(object_key).put()
logger.info("HTTPStatusCode: %s", response['ResponseMetadata']['HTTPStatusCode'])
logger.info("RequestId: %s", response['ResponseMetadata']['RequestId'])
logger.info("HostId: %s", response['ResponseMetadata']['HostId'])
logger.info("Date: %s", response['ResponseMetadata']['HTTPHeaders']['date'])
```

Sie können auch Ausnahmen abfangen und relevante Informationen protokollieren, wenn eine Ausnahme ausgelöst wird. Weitere Informationen finden Sie unter [Erkennen nützlicher Informationen anhand von Fehlerantworten](#) in der AWS SDKfor Python (Boto) -Referenz. API

Darüber hinaus können Sie Boto3 mit dem folgenden Code für die Ausgabe ausführlicher Debugging-Protokolle konfigurieren:

```
import boto3
boto3.set_stream_logger('', logging.DEBUG)
```

Weitere Informationen finden Sie [set_stream_logger](#) in der APIReferenz AWS SDK für Python (Boto).

Verwenden von SDK for Ruby, um eine Anfrage zu erhalten IDs

Sie können Ihre Anfrage IDs mit den SDK for Ruby-Versionen 1, 2 oder 3 erhalten.

- Verwenden von SDK for Ruby — Version 1 — Sie können HTTP Wire Logging global mit der folgenden Codezeile aktivieren.

```
s3 = AWS::S3.new(:logger => Logger.new($stdout), :http_wire_trace => true)
```

- Verwenden Sie SDK for Ruby — Version 2 oder Version 3 — Sie können HTTP Wire Logging global mit der folgenden Codezeile aktivieren.

```
s3 = Aws::S3::Client.new(:logger => Logger.new($stdout), :http_wire_trace => true)
```

Tipps zum Abrufen von Wire-Informationen von einem AWS Client finden Sie unter [Debug-Tipp: Wire-Trace-Informationen von einem Client abrufen](#).

Verwenden von, um AWS CLI eine Anfrage zu erhalten IDs

Um Ihre Anfrage zu erhalten IDs, wenn Sie die AWS Command Line Interface (AWS CLI) verwenden, fügen Sie `--debug` sie zu Ihrem Befehl hinzu.

Verwenden Sie Windows PowerShell , um die Anfrage abzurufen IDs

Informationen zum Wiederherstellen von Protokollen mit Windows PowerShell finden Sie unter [Response Logging in AWS Tools for Windows PowerShell](#). NETBlogbeitrag zur Entwicklung.

Verwenden von AWS CloudTrail Datenereignissen zum Abrufen von Anfragen IDs

Ein Amazon S3-Bucket, der mit CloudTrail Datenereignissen konfiguriert ist, um API S3-Operationen auf Objektebene zu protokollieren, bietet detaillierte Informationen über Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon S3 ausgeführt werden. Sie können eine [S3-Anfrage identifizieren, IDs indem Sie CloudTrail Ereignisse mit Athena abfragen](#).

Verwenden Sie die S3-Serverzugriffsprotokollierung, um eine Anfrage abzurufen IDs

Ein Amazon-S3-Bucket, der für die S3-Server-Zugriffsprotokollierung konfiguriert ist, stellt detaillierte Aufzeichnungen zu allen an den Bucket gesendeten Anforderungen bereit. Sie können S3-Anfragen identifizieren, IDs indem Sie [die Serverzugriffsprotokolle mit Athena abfragen](#).

Dokumentverlauf

- Aktuelle API Version: 2006-03-01

In der folgenden Tabelle werden die wichtigen Änderungen in den einzelnen Versionen der Amazon Simple Storage Service API Reference und des Amazon S3 S3-Benutzerhandbuchs beschrieben. Um über Aktualisierungen dieser Dokumentation informiert zu werden, können Sie einen RSS Feed abonnieren.

Änderung	Beschreibung	Datum
Neues Standardverhalten bei der Umstellung auf minimale Objektgrößen für Amazon S3 Lifecycle-Konfigurationen.	Amazon S3 wendet jetzt ein Standardverhalten auf S3-Lifecycle-Konfigurationen an, das verhindert, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden. Informationen zum Überschreiben dieses Verhaltens finden Sie unter Zulassen der Übertragung von Objekten, die kleiner als 128 KB sind.	24. September 2024
S3 Access Grants unterstützt eine API Aktion, die die Zugriffsberechtigungen des Anrufers auflistet.	IAM-Identitäten und IAM Identity Center-Unternehmensverzeichnisidentitäten können jetzt die verwenden <code>ListCallerAccessGrants</code> API um alle Amazon S3 S3-Buckets, -Präfixe und Objekte aufzulisten, auf die sie zugreifen können, wie in ihren S3 Access Grants definiert. Verwenden Sie diese OptionAPI, um alle S3-	5. September 2024

Daten zu ermitteln, auf die eine IAM oder die Identität des Unternehmensverzeichnisses über S3 Access Grants innerhalb eines bestimmten Zeitraums zugreifen kann AWS-Konto. Weitere Informationen finden Sie unter [Auflisten der Zugriffsberechtigungen des Anrufers](#).

[Amazon S3 unterstützt die Verwendung von bedingten Schreibvorgängen für PutObject und CompleteMultipartUpload](#)

Sie können überprüfen, ob ein Objekt in Ihrem Bucket vorhanden ist, bevor Sie es mithilfe von bedingten Schreibvorgängen beim Hochladen erstellen. Dadurch kann das Überschreiben vorhandener Daten verhindert werden. Durch bedingte Schreibvorgänge wird bestätigt, dass sich in Ihrem Bucket noch kein Objekt mit demselben Schlüsselnamen befindet. Weitere Informationen finden Sie unter [Bedingte Anfragen](#).

20. August 2024

[Amazon S3 Select ist für Neukunden nicht mehr verfügbar](#)

Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. [Weitere Informationen](#).

25. Juli 2024

Amazon S3 Inventory unterstützt den InventoryAccessibl eOptionalFields Bedingung sschlüssel s3:	Amazon S3 Inventory unterstützt den InventoryAccessibl eOptionalFields Bedingung sschlüssel s3: zur Steuerung , ob Benutzer optionale Metadatenfelder in ihre Berichte aufnehmen können. Weitere Informationen finden Sie unter Steuern der Erstellung g der Konfiguration von S3- Inventarberichten .	20. Februar 2024
IPv6Unterstützung für S3 auf Outposts	Sie können jetzt auf S3 in Outposts-Buckets zugreifen , indem Sie IPv6 über S3 auf Outposts-Dual-Stack-Endpunk ten auf Outposts zugreifen . IPv6Die Unterstützung für S3 on Outposts ermöglicht es Ihnen, Ihre S3 on Outposts- Buckets zu verwalten und Ebenenressourcen über Netzwerke zu steuern. IPv6	16. Januar 2024
Neue leistungsstarke Amazon- S3-Speicherklasse mit einer Zone – S3 Express One Zone	Amazon S3 Express One Zone ist eine leistungsstarke Amazon-S3-Speicherklasse mit einer einzelnen Zone, die speziell für den konsisten ten Datenzugriff im einstell igen Millisekundenbereich für Ihre latenzempfindlichs ten Anwendungen entwickelt wurde. Weitere Informationen finden Sie unter S3 Express One Zone .	28. November 2023

[Mountpoint für Amazon S3
fügt Unterstützung für S3
Express One Zone hinzu](#)

Sie können jetzt S3-Express-One-Zone-Verzeichnis-Buckets mit [Mountpoint](#) mounten.

28. November 2023

[Lambda-Aufrufschema-version](#)

Amazon S3 Batch Operations führt eine neue Lambda-Aufrufschema-version für Batch-Operations-Aufträge ein, die auf Verzeichnis-Buckets ausgeführt werden. Verwenden von Lambda und Amazon S3 Batch Operations mit Verzeichnis-Buckets.

28. November 2023

[Importaktion für Verzeichnis-
Buckets](#)

Amazon S3 führt die Importaktion ein. Import ist eine optimierte Methode zur Erstellung von Amazon-S3-Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets. Weitere Informationen finden Sie unter [Importieren von Objekten in einen Verzeichnis-Bucket](#).

28. November 2023

[Verwalten des S3-Zugriffs mit S3 Access Grants](#)

Mit Amazon S3 Access Grants können Sie Datenberechtigungen in großem Umfang verwalten für AWS Identity and Access Management (IAM) Prinzipale zusätzlich zu Verzeichnisidentitäten aus Unternehmensverzeichnissen wie Azure AD. Sie können jetzt S3-Berechtigungen mit den geringsten Rechten durchsetzen und diese Berechtigungen ganz einfach an Ihre Geschäftsanforderungen anpassen. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit S3 Access Grants](#).

26. November 2023

[Mountpoint für Amazon S3 fügt Caching-Feature hinzu](#)

Mit [Mountpoint](#) können Sie jetzt das Caching für Daten konfigurieren, auf die wiederholt zugegriffen wird.

22. November 2023

[Verbesserte Amazon S3 Batch Operations-Manifestgenerierung](#)

Sie können Amazon S3 Batch Operations jetzt anweisen, automatisch ein Manifest auf der Grundlage von Objektkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3 S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mit dem AWS CLI, AWS SDKs, oder Amazon S3 RESTAPI. Weitere Informationen finden Sie unter [Erstellen eines Amazon-S3-Batch-Operations-Auftrags](#).

22. November 2023

[Bestehende Amazon-S3-Buckets können jetzt Object-Lock-Konfigurationen hinzufügen](#)

Sie können Object Lock jetzt für einen vorhandenen Amazon-S3-Bucket aktivieren. Sie können gesetzliche und andere Aufbewahrungsfristen für neue oder bestehende Buckets festlegen. Weitere Informationen finden Sie unter [Verwenden von Object Lock](#).

20. November 2023

[S3-Storage-Lens-Anforderungsmetriken für Präfixe](#)

In S3 Storage Lens werden Anforderungsmetriken für Präfixe in einem Amazon-S3-Bucket eingeführt. Weitere Informationen finden Sie unter [Metrikkategorien](#).

17. November 2023

[Amazon-S3-Storage-Lens-Gruppen](#)

In S3 Storage Lens werden Storage-Lens-Gruppen eingeführt. Hierbei handelt es sich um benutzerdefinierte Filter für Objekte auf der Grundlage von Objektmetadaten. Weitere Informationen finden Sie unter [Arbeiten mit S3-Storage-Lens-Gruppen](#).

15. November 2023

[Neue IAM Richtlinie](#)

S3 on Outposts führt die serviceverknüpfte Rolle `AWSServiceRoleForS3onOutposts` ein, die Sie bei der Verwaltung von Netzwerkressourcen unterstützt. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für S3 on Outposts](#).

3. Oktober 2023

[Amazon S3 bietet den Last-Modified -Zeitpunkt für das Löschen von Markierungen](#)

Amazon S3 gibt die Last-Modified Uhrzeit der Löschmarkierungen in den Antwort-Headern von S3 Head und Get API Vorgängen an. Weitere Informationen finden Sie unter [Arbeiten mit Löschmarkierungen](#).

27. September 2023

[Amazon S3 S3-Update auf AWS verwaltete Richtlinie](#)

Amazon S3 hat die `s3:Describe*` -Berechtigungen zu `AmazonS3ReadOnlyAccess` hinzugefügt. Weitere Informationen finden Sie unter [AWS verwaltete Richtlinien für Amazon S3](#).

11. August 2023

[Verbesserte Startzeiten für Standard-Wiederherstellungsanfragen, die über S3 Batch Operations gestellt wurden](#)

Standardabrufe für Wiederherstellungsanfragen, die über S3 Batch Operations gestellt wurden, können jetzt innerhalb von Minuten beginnen. Weitere Informationen finden Sie unter [Optionen für den Archivabruf](#).

9. August 2023

[Mountpoint, ein Client mit hohem Durchsatz zum Mounten eines Amazon-S3-Buckets als lokales Dateisystem wurde hinzugefügt.](#)

Mit [Mountpoint](#) können Ihre Anwendungen über Dateioperationen auf die in Amazon S3 gespeicherten Objekte zugreifen, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und den Durchsatz von Amazon S3 zugreifen können.

9. August 2023

[Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service Schlüssel \(-\) DSSE KMS](#)

Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) keys (DSSE-KMS) wendet zwei Verschlüsselungsebenen auf Objekte an, wenn sie auf Amazon S3 hochgeladen werden. Weitere Informationen finden Sie unter [Verwenden der serverseitigen Dual-Layer-Verschlüsselung mit AWS KMS Schlüssel](#).

13. Juni 2023

[Amazon S3 aktiviert S3 Block Public Access und deaktiviert S3-Zugriffskontrolllisten \(ACLs\) für alle neuen Buckets.](#)

Amazon S3 aktiviert jetzt automatisch S3 Block Public Access und deaktiviert S3-Zugriffskontrolllisten (ACLs) für alle neuen S3-Buckets insgesamt AWS Regionen. Weitere Informationen finden Sie unter [Sperrungen des öffentlichen Zugriffs auf Ihren Amazon S3 S3-Speicher](#) und [Steuern des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

27. April 2023

[Metrik zu fehlgeschlagenen S3-Replikationsvorgängen](#)

Amazon S3 fügt neue hinzu Amazon CloudWatch Metrik zur Überwachung von S3-Replikationsfehlern. Weitere Informationen finden Sie unter [Überwachen des Fortschritts mit Replikationsmetriken](#).

05. April 2023

Privat DNS	AWS PrivateLink für Amazon S3 unterstützt jetzt PrivateDNS. Weitere Informationen finden Sie unter Privat DNS .	14. März 2023
Support für kontoübergreifende Zugriffspunkte in der Amazon-S3-Konsole	Amazon S3 unterstützt jetzt die Erstellung von kontoübergreifenden Zugriffspunkten mit der Amazon-S3-Konsole. Weitere Informationen finden Sie unter Erstellen von Zugriffspunkten .	14. März 2023
Amazon S3 in Outposts unterstützt S3 Replication in Outposts	Mit der lokalen S3-Replikation können Sie Objekte automatisch in einen einzelnen Ziel-Bucket oder in mehrere Ziel-Buckets von Outposts replizieren. Die Ziel-Buckets können unterschiedlich sein AWS Outposts oder innerhalb derselben Outposts wie der Quell-Bucket. Weitere Informationen finden Sie unter Replikation von Objekten für S3 in Outposts .	14. März 2023

[Access-Point-Alias in Amazon S3 Object Lambda](#)

Wenn Sie einen Object Lambda Access Point erstellen , generiert Amazon S3 automatisch einen eindeutigen Alias für den Object Lambda Access Point. Sie können diesen Alias anstelle eines Amazon S3 S3-Bucket-Namens oder des Object Lambda Access Point Amazon Resource Name (ARN) in einer Anfrage für Access Point-Datenebenenoperationen verwenden. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren Object Lambda Access Point](#).

14. März 2023

[Kontoübergreifender Support für Multi-Region Access Points in Amazon S3](#)

Amazon S3 unterstützt jetzt die Erstellung von kontoübergreifenden Multi-Region Access Points mit der Amazon-S3-Konsole. Weitere Informationen finden Sie unter [Erstellen von Multi-Region Access Points](#).

14. März 2023

[Kontoübergreifende Zugriffspunkte](#)

Amazon S3 unterstützt das Erstellen von kontoübergreifenden Zugriffspunkten. Sie können einen kontoübergreifenden Access Point erstellen, indem Sie den AWS Command Line Interface (AWS CLI) oder die REST API `CreateAccessPoint` Operation. Weitere Informationen finden Sie unter [Erstellen von Zugriffspunkten](#).

30. November 2022

[Amazon S3 unterstützt Failover-Kontrollen für Amazon S3 Multi-Region Access Points](#)

Amazon S3 führt Failover-Kontrollen für Multi-Region Access Points ein. Mit diesen Steuerelementen können Sie den Datenverkehr von S3-Datenzugriffsanfragen, der über einen Amazon S3 Multiregion Access Point geleitet wird, auf einen anderen übertragen. AWS-Region innerhalb weniger Minuten, um hochverfügbare Anwendungen zu testen und zu erstellen. Weitere Informationen finden Sie unter [Failover-Kontrollen für Amazon S3 Multi-Region Access Points](#).

28. November 2022

[Amazon S3 Storage Lens erhöht die unternehmensweite Sichtbarkeit mit 34 neuen Metriken](#)

S3 Storage Lens führt 34 zusätzliche Metriken ein, um umfassende Möglichkeiten zur Kostenoptimierung aufzudecken, bewährte Methoden für den Datenschutz zu identifizieren und die Leistung von Anwendungsworkflows zu verbessern. Weitere Informationen finden Sie unter [Metriken von S3 Storage Lens](#).

17. November 2022

[Amazon S3 unterstützt höhere Geschwindigkeiten für Wiederherstellungsanforderungen für S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive](#)

Amazon S3 unterstützt Wiederherstellungsanfragen mit einer Geschwindigkeit von bis zu 1.000 Transaktionen pro Sekunde AWS-Konto für die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive.

15. November 2022

[Amazon S3 on Outposts unterstützt zusätzliche S3-Lebenszyklusaktionen und -filter](#)

S3 on Outposts unterstützt zusätzliche S3-Lebenszyklusregeln zur Optimierung des Kapazitätsmanagements. Sie können Objekte ablaufen lassen, wenn sie veralten oder durch neuere Versionen ersetzt werden. Sie können eine Lebenszyklusregel für einen ganzen Bucket oder eine Teilmenge von Objekten in einem Bucket erstellen, indem Sie nach Präfixen, Objekt-Tags oder Objektgröße filtern. Weitere Informationen finden Sie unter [Erstellen und Verwalten einer Lebenszyklus-Konfiguration](#).

2. November 2022

[S3-Replikationsunterstützung für SSE -C-Objekte](#)

Sie können Objekte replizieren, die mit der serverseitigen Verschlüsselung unter Verwendung der vom Kunden bereitgestellten Schlüssel erstellt wurden. Weitere Hinweise zum Replizieren verschlüsselter Objekte finden Sie unter [Replizieren von Objekten, die mit serverseitiger Verschlüsselung \(SSE-C, -S3, SSE -\) erstellt wurden](#).
SSE KMS

24. Oktober 2022

[Amazon S3 on Outposts unterstützt Zugriffspunkt-Aliasse](#)

Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Jedes Mal, wenn Sie einen Zugriffspunkt für einen Bucket erstellen, generiert S3 on Outposts automatisch einen Zugriffspunkt-Alias. Sie können diesen Zugriffspunkt-Alias anstelle eines Zugriffspunkts ARN für jeden Vorgang auf der Datenebene verwenden. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets](#).

21. Oktober 2022

[S3-Objekt-Lambda unterstützt die Operationen HeadObject, ListObjects und ListObjectsV2](#)

Sie können benutzerdefinierte Lambda Code verwenden, um die von den S3-Standardanforderungen GET, LIST oder HEAD zurückgegebenen Daten zu ändern, um Zeilen zu filtern, die Größe von Images dynamisch anzupassen, vertrauliche Daten zu unterdrücken und vieles mehr. Weitere Informationen finden Sie unter [Transformieren von Objekten mit S3-Objekt Lambda](#).

4. Oktober 2022

[Amazon S3 on Outposts unterstützt die S3-Versionsverwaltung](#)

Wenn diese Option aktiviert ist, speichert die S3-Versionsverwaltung mehrere unterschiedliche Kopien eines Objekts im selben Bucket. Sie können die S3-Versionsverwaltung verwenden, um sämtliche Versionen aller Objekte in Ihren Outposts-Buckets zu speichern, abzurufen oder wiederherzustellen. Mit der S3-Versionsverwaltung können Sie Versionen sowohl nach unbeabsichtigten Benutzeraktionen als auch nach Anwendungsfehlern problemlos wiederherstellen. Weitere Informationen finden Sie unter [Verwalten der S3-Versionsverwaltung für Ihren S3-on-Outposts-Bucket](#).

21. September 2022

[AWS Backup für Amazon S3](#)

AWS Backup ist ein vollständig verwalteter, richtlinienbasierter Service, mit dem Sie eine zentrale Backup-Richtlinie zum Schutz Ihrer Amazon S3 S3-Daten definieren können. Weitere Informationen finden Sie unter [Verwenden AWS Backup für Amazon S3](#).

18. Februar 2022

[Verwenden der S3-Batchreplikation, um bestehende Objekte zu replizieren](#)

Mit der S3-Batchreplikation können Sie Objekte replizieren, die vor der Einführung einer Replikationskonfiguration existierten. Das Replizieren vorhandener Objekte erfolgt mithilfe eines Auftrags für Batchoperationen. Die S3-Batchreplikation unterscheidet sich von der Live-Replikation, die kontinuierlich und automatisch neue Objekte über Amazon-S3-Buckets hinweg kopiert. Weitere Informationen finden Sie unter [Replizieren vorhandener Objekte mit S3-Batch-Replikation](#).

8. Februar 2022

[Umbenennen von S3 Glacier Flexible Retrieval](#)

Die Glacier-Speicherklasse wurde in S3 Glacier Flexible Retrieval umbenannt. Diese Änderung hat keine Auswirkungen auf die API.

30. November 2021

[Neue Einstellung für S3-Objekt
eigentümer zum Deaktivieren
ACLs](#)

Sie können die vom Bucket Ownership erzwungene Einstellung für Object Ownership anwenden, um sie ACLs für Ihren Bucket und die darin enthaltenen Objekte zu deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket zu übernehmen. Die vom Bucket-Eigentümer erzwungene Einstellung vereinfacht die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten. Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket](#).

30. November 2021

[Neue S3-Intelligent-Tiering-
Speicherklasse](#)

S3 Intelligent-Tiering Archive Instant Access ist eine zusätzliche Speicherklasse unter S3 Intelligent-Tiering. Weitere Informationen finden Sie unter [Wie funktioniert S3 Intelligent-Tiering](#).

30. November 2021

[Neue Speicherklasse S3
Glacier Instant Retrieval](#)

Sie können jetzt Objekte in der Speicherklasse S3 Glacier Instant Retrieval platzieren. Weitere Informationen zu dieser Speicherklasse finden Sie unter [Verwenden von Amazon-S3-Speicherklassen](#).

30. November 2021

[AWS Backup für Amazon S3 Preview](#)

AWS Backup ist ein vollständig verwalteter, richtlinienbasierter Service, mit dem Sie eine zentrale Backup-Richtlinie zum Schutz Ihrer Amazon S3 S3-Daten definieren können. Weitere Informationen finden Sie unter [Verwenden AWS Backup für Amazon S3](#).

30. November 2021

[AWS Identity and Access Management Access Analyzer für Amazon S3](#)

IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM Richtliniengrammatik und der bewährten Methoden zu validieren. Weitere Informationen zur Überprüfung von Richtlinien mit IAM Access Analyzer finden Sie unter [IAMAccess Analyzer-Richtlinieninvalidierung](#) im IAMBenutzerhandbuch.

30. November 2021

[Neue Ereignistypen](#)

Zu Amazon-S3-Ereignisbenachrichtigungen hinzugefügte Ereignistypen finden Sie unter [Amazon-S3-Ereignisbenachrichtigungen](#).

29. November 2021

[Amazon für EventBridge Buckets aktivieren](#)

Sie können EventBridge auf Amazon S3 S3-Bucket das Senden von Ereignissen an Amazon aktivieren EventBridge, siehe [Verwenden EventBridge](#).

29. November 2021

[Neue S3-Lebenszyklusfilter](#)

Sie können Lebenszyklusregeln basierend auf der Objektgröße erstellen oder angeben, wie viele nicht aktuelle Objektversionen beibehalten werden sollen. Weitere Informationen finden Sie unter [Beispiele der S3-Lebenszyklus-Konfiguration](#).

23. November 2021

[Veröffentlichen Sie Amazon S3 Storage Lens-Metriken auf Amazon CloudWatch](#)

Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch Dashboards zu erstellen. Sie können auch CloudWatch Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um die S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Darüber hinaus CloudWatch APIs ermöglichen sie Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3 Storage Lens-Metriken. Weitere Informationen finden Sie unter [Messdaten zur Überwachung von S3 Storage Lens unter CloudWatch](#).

22. November 2021

Multiregionale Zugriffspunkte

Sie können Multi-Region-Access Points verwenden, um einen globalen Endpunkt zu erstellen, über den Anwendungen Anfragen von Amazon S3 S3-Buckets bearbeiten können, die sich in mehreren befinden. AWS-Regionen. Sie können diesen Access Point für mehrere Regionen verwenden, um Daten an einen Bucket mit der geringsten Latenz weiterzuleiten. Weitere Informationen zu Multi-Regions-Zugriffspunkten und deren Verwendung finden Sie unter [Multi-Regions-Zugriffspunkte in Amazon S3](#).

2. September 2021

[Amazon S3 in Outposts fügt direkten lokalen Zugriff für Anwendungen hinzu](#)

Führen Sie Ihre Anwendungen außerhalb der AWS Outposts virtuelle private Cloud (VPC) und greifen Sie auf Ihre S3 on Outposts-Daten zu. Sie können auch direkt aus Ihrem lokalen Netzwerk auf S3 in Outposts zugreifen. Weitere Informationen zur Konfiguration von S3 auf Outposts-Endpunkten mit [kundeneigenen IP-Adressen \(CoIP\)](#) und zum Zugreifen auf Ihre Objekte, indem Sie ein [lokales Gateway von Ihrem lokalen Netzwerk](#) aus einrichten, finden Sie unter [Zugreifen auf Amazon S3 auf Outposts](#) mit Nur-Zugriffspunkten. VPC

29. Juli 2021

[Amazon-S3-Zugriffspunktalias](#)

Wenn Sie einen Zugriffspunkt erstellen, generiert Amazon S3 automatisch einen Alias, den Sie anstelle eines Bucket-Namens für den Datenzugriff verwenden können. Sie können diesen Access Point-Alias anstelle eines Amazon-Ressourcennamens (ARN) für jeden Access Point-Datenebenenvorgang verwenden. Weitere Informationen finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren Zugriffspunkt](#).

26. Juli 2021

[Amazon S3 Inventory und S3 Batch Operations unterstützen den Status des S3-Bucket-Schlüssels](#)

Amazon-S3-Bestand und Batch-Vorgänge unterstützen das Identifizieren und Kopieren vorhandener Objekte mit S3-Bucket-Schlüssel. S3-Bucket-Schlüssel beschleunigen die Preissenkung der serverseitigen Verschlüsselung für vorhandene Objekte. Weitere Informationen finden Sie unter [Amazon S3 Inventory](#) und [Batch Operations Copy object \(Batch-Vorgänge Objekt kopieren\)](#).

3. Juni 2021

[Amazon S3 Storage-Lens Metriken Konto-Snapshot](#)

Der S3-Storage-Lens-Konto-Snapshot zeigt Ihren Gesamtspeicher, die Objektzahl und die durchschnittliche Objektgröße auf der Startseite der S3-Konsole (Buckets) an, indem Metriken aus Ihrem Standard-Dashboard zusammengefasst werden. Weitere Informationen finden Sie unter [S3 Storage Lens Metrics Konto-Snapshot](#).

5. Mai 2021

[Erhöhte Unterstützung von Amazon-S3-in-Outposts-Endpunkten](#)

S3 in Outposts unterstützt jetzt bis zu 100 Endpunkte pro Outpost. Weitere Informationen finden Sie unter [S3-in-Outposts-Netzeinschränkungen](#).

29. April 2021

[Amazon S3 on Outposts — Ereignisbenachrichtigungen in Amazon CloudWatch Events](#)

Sie können CloudWatch Ereignisse verwenden, um eine Regel zu erstellen, mit der alle S3 on API Outposts-Ereignisse erfasst und über alle unterstützten CloudWatch Ziele benachrichtigt werden. Weitere Informationen finden Sie unter [Empfangen von S3 on Outposts-Ereignisbenachrichtigungen mithilfe von CloudWatch Ereignissen](#).

19. April 2021

[S3 Objekt Lambda](#)

Mit S3 Object Lambda können Sie GET-Anforderungen von Amazon S3 eigenen Code hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Sie können mit benutzerdefiniertem Code die von standardmäßigen S3-GET-Anforderungen zurückgegebenen Daten ändern, um u. a. Zeilen zu filtern, Bilder dynamisch in der Größe zu ändern und vertrauliche Daten zu redigieren und mehr. Weitere Informationen finden Sie unter [Transformieren von Objekten](#).

18. März 2021

[AWS PrivateLink](#)

Mit AWS PrivateLink Für Amazon S3 können Sie eine direkte Verbindung zu S3 herstellen, indem Sie einen Schnittstellenendpunkt in Ihrer virtuellen privaten Cloud (VPC) verwenden, anstatt eine Verbindung über das Internet herzustellen. Auf Schnittstellen-Endpunkte kann direkt von Anwendungen aus zugegriffen werden, die sich vor Ort oder an einem anderen Standort befinden. Weitere Informationen finden Sie unter [.AWS PrivateLink für Amazon S3.](#)

2. Februar 2021

[Verwaltung der Kapazität von Amazon S3 on Outposts mit AWS CloudTrail](#)

Verwaltungsereignisse von S3 on Outposts sind in CloudTrail Protokollen verfügbar. Weitere Informationen finden Sie unter [Verwaltung der Kapazität von S3 on Outposts mit CloudTrail.](#)

21. Dezember 2020

Starke Konsistenz

Amazon S3 bietet insgesamt eine hohe read-after-write Konsistenz für Objekte PUT und DELETE Anfragen von Objekten in Ihrem S3-Bucket AWS-Regionen. Darüber hinaus sind die Lesevorgänge auf Amazon S3 Select, Amazon S3 S3-Zugriffskontrolllisten, Amazon S3 S3-Objekt-Tags und Objektmetadaten (z. B. HEAD Objekt) sehr konsistent. Weitere Informationen finden Sie unter [Amazon-S3-Datenkonsistenzmodell](#).

1. Dezember 2020

Synchronisierung der Amazon-S3-Replikatänderung

Die Amazon S3 S3-Replikationsmodifikationssynchronisierung sorgt dafür, dass Objektmetadaten wie Tags und Object Lock-Einstellungen zwischen Quellobjekten und Replikaten synchron bleiben. ACLs Wenn diese Funktion aktiviert ist, repliziert Amazon S3 Metadatenänderungen, die entweder am Quellobjekt oder an den Replikatkopien vorgenommen wurden. Weitere Informationen finden Sie unter [Replizieren von Metadatenänderungen mit der Synchronisierung von Replikatänderungen](#).

1. Dezember 2020

Amazon-S3-Bucket-Schlüssel

Amazon S3 Bucket Keys reduzieren die Kosten für die serverseitige Amazon S3 S3-Verschlüsselung mit AWS Key Management Service (SSE-KMS). Dieser neue Schlüssel auf Bucket-Level für serverseitige Verschlüsselung kann Folgendes reduzieren AWS KMS Anforderungskosten um bis zu 99 Prozent durch Verringerung des Anforderungsverkehrs von Amazon S3 auf AWS KMS. Weitere Informationen finden Sie unter [Reduzierung der Kosten von SSE — mithilfe KMS von S3 Bucket Keys](#).

1. Dezember 2020

[Amazon-S3-Storage-Lens](#)

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Ihr Dashboard bietet Drilldown-Optionen zur Generierung und Visualisierung von Erkenntnissen über die Organisation, den Account, AWS-Region, Speicherklasse, Bucket, Präfix oder Storage Lens auf Gruppenebene. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren. Weitere Informationen finden Sie unter [Bewertung Ihrer Speicherkapazität und -nutzung mit S3 Storage Lens](#).

18. November 2020

[Verfolgen von S3-Anfragen mit AWS X-Ray](#)

Amazon S3 lässt sich in X-Ray integrieren, um den [Trace-Kontext](#) zu verbreiten und Ihnen eine Anforderungskette mit [Upstream- und Downstream-Knoten](#) bereitzustellen. Weitere Informationen finden Sie unter [Verfolgen von Anforderungen mit X-Ray](#).

16. November 2020

[S3-Replikationsmetriken](#)

S3-Replikationsmetriken enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Weitere Informationen finden Sie unter [Replikationsmetriken und Amazon-S3-Ereignis-Benachrichtigungen](#).

9. November 2020

[S3-Intelligent-Tiering-Archivzugriff und tiefer Archivzugriff](#)

S3-Intelligent-Tiering-Archivzugriff und tiefer Archivzugriff sind zusätzliche Speicherebenen unter S3-Intelligent-Tiering. Weitere Informationen finden Sie unter [Speicherklassen zum automatischen Optimieren häufig und selten aufgerufener Objekte](#).

9. November 2020

[Löschmarkierungsreplikation](#)

Mit der Replikation von Löschmarkierungen können Sie sicherstellen, dass Löschmarkierungen für Ihre Replikationsregeln in Ihre Ziel-Buckets kopiert werden. Weitere Informationen finden Sie unter [Verwenden der Löschmarkierungs-Replikation](#).

9. November 2020

[S3 Object Ownership](#)

Object Ownership ist eine S3-Bucket-Einstellung, mit der Sie die Eigentümerschaft an neuen Objekten, die in Ihre Buckets hochgeladen werden, steuern können. Weitere Informationen finden Sie unter [Verwenden von S3-Objekt-Eigentümerschaft](#).

2. Oktober 2020

[Amazon S3 in Outposts](#)

Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihrem erstellten AWS Outposts Ressourcen und einfaches Speichern und Abrufen von Objekten vor Ort für Anwendungen, die lokalen Datenzugriff, lokale Datenverarbeitung und Datenresidenz erfordern. Sie können S3 auf Outposts verwenden über AWS Management Console, AWS CLI, AWS SDKs, oder RESTAPI. Weitere Informationen finden Sie unter [Verwenden von Amazon S3 in Outposts](#).

30. September 2020

[Bedingung „Bucket-Eigentümer“](#)

Sie können die Bedingung Amazon S3 S3-Bucket-Besitzer verwenden, um sicherzustellen, dass die Buckets, die Sie in Ihren S3-Vorgängen verwenden, zu den AWS-Konten das erwarten Sie. Weitere Informationen finden Sie unter [Bucket-Eigentümer-Bedingung](#).

11. September 2020

[Unterstützung von S3-Batch-Vorgang für die Aufbewahrung von Objektsperren](#)

Sie können jetzt Batch-Vorgänge mit der S3-Objektsperre verwenden, um Aufbewahrungseinstellungen auf viele Amazon-S3-Objekte gleichzeitig anzuwenden. Weitere Informationen finden Sie unter [Festlegen von Aufbewahrungsfristen für die S3-Objektsperre mit S3-Batchvorgängen](#).

4. Mai 2020

[Unterstützung von S3-Batchoperationen für die gesetzliche Aufbewahrungspflicht von Objektsperren](#)

Sie können jetzt Batchvorgänge mit der S3-Objektsperre verwenden, um vielen Amazon-S3-Objekten gleichzeitig eine gesetzliche Aufbewahrungsfrist zuzuweisen. Weitere Informationen finden Sie unter [Verwenden von S3-Batch-Vorgänge zum Festlegen von gesetzlichlichen Aufbewahrungsfristen für die S3-Objektsperre](#).

4. Mai 2020

[Aufgaben-Tags für S3-Batchvorgängen](#)

Sie können Ihren S3-Batchvorgängeaufträgen Markierungen hinzufügen, um diese Aufträge zu steuern und zu kennzeichnen. Weitere Informationen finden Sie unter [Tags für S3-Batchoperationenaufträge](#).

16. März 2020

[Amazon-S3-Zugriffspunkte](#)

Amazon-S3-Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie S3-Objekt-Vorgänge ausführen können. Weitere Informationen finden Sie unter [Verwalten des Datenzugriffs mit Amazon-S3-Zugriffspunkten](#).

02. Dezember 2019

[Zugriffs-Analyzer für Amazon S3](#)

Access Analyzer for Amazon S3 warnt Sie vor S3-Buckets, die so konfiguriert sind, dass sie jedem Zugriff auf das Internet oder andere Benutzer ermöglichen AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation. Weitere Informationen finden Sie unter [Verwenden des Zugriffs-Analyzer für Amazon S3](#).

02. Dezember 2019

[S3-Kontrolle der Replikationszeit \(S3RTC\)](#)

S3 Replication Time Control (S3RTC) repliziert die meisten Objekte, die Sie auf Amazon S3 hochladen, innerhalb von Sekunden und 99,99 Prozent dieser Objekte innerhalb von 15 Minuten. Weitere Informationen finden Sie unter [Objekte mit S3 Replication Time Control \(S3\) replizieren](#). RTC

20. November 2019

[Same-Region Replication \(SRR, Replikation in derselben Region\)](#)

Sie können Same-Region Replication (SRR) verwenden, um Objekte zwischen Amazon S3 S3-Buckets im selben zu kopieren AWS-Region. [Informationen sowohl zur regionsübergreifenden Replikation \(CRR\) als auch zur regionsübergreifenden Replikation finden Sie unter Replikation](#).

18. September 2019

[Unterstützung der regionsübergreifenden Replikation für die S3-Objektsperre](#)

Die regionsübergreifende Replikation unterstützt jetzt Objektsperren. Weitere Informationen finden Sie unter [Was repliziert Amazon S3?](#).

28. Mai 2019

[S3-Batch-Vorgänge](#)

Mit S3-Batchvorgängen können Sie groß angelegte Batchvorgänge für Amazon-S3-Objekte durchführen. S3-Batch-Vorgänge kann eine einzelne Operation für Listen von Objekten ausführen, die Sie angeben. Ein einziger Auftrag kann die festgelegte Operation auf Milliarden von Objekten mit mehreren Exabytes an Daten durchführen. Weitere Informationen finden Sie unter [Durchführen von S3-Batch-Vorgänge](#).

30. April 2019

[Region Asien-Pazifik \(Hongkong\)](#)

Amazon S3 ist jetzt in der Region Asien-Pazifik (Hongkong) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

24. April 2019

[Hinzufügung eines neuen Feldes zu den Serverzugriffsprotokollen](#)

Amazon S3 hat den Serverzugriffsprotokollen das folgende neue Feld hinzugefügt: Transport Layer Security (TLS)-Version. Weitere Informationen finden Sie unter [Server-Zugriffsprotokollformat](#).

28. März 2019

[Neue Archivspeicherklasse](#)

Amazon S3 bietet eine neue Archivspeicherklasse, S3 Glacier Deep Archive (DEEP_ARCHIVE), zum Speichern selten benötigter Objekte. Weitere Informationen finden Sie unter [Speicherklassen](#).

27. März 2019

[Hinzufügung neuer Felder zu den Serverzugriffsprotokollen](#)

In Amazon S3 wurden die folgenden neuen Felder zu den Serverzugriffs-Protokollen hinzugefügt: Host-ID, Signature Version (Signature Version), Cipher Suite, Authentication Type (Authentifizierungstyp) und Host Header. Weitere Informationen finden Sie unter [Server-Zugriffsprotokollformat](#).

5. März 2019

[Unterstützung für Amazon-S3-Inventory-Dateien im Parquet-Format](#)

Amazon S3 unterstützt jetzt das [Apache Parquet-Format \(Parquet\)](#) zusätzlich zu den von [Apache optimierten Dateiformaten für Zeilen, Spalten \(ORC\)](#) und kommagetrennte Werte (CSV) für Inventarausgabedateien. Weitere Informationen finden Sie unter [Bestand](#).

4. Dezember 2018

[S3-Objektsperre](#)

Amazon S3 bietet jetzt Object Lock-Funktionen, die Write Once Read Many (WORM)-Schutz für Amazon S3 S3-Objekte bieten. Weitere Informationen finden Sie unter [Sperrern von Objekten](#).

26. November 2018

[Wiederherstellungsgeschwindigkeit-Upgrade](#)

Mit dem Amazon-S3-Wiederherstellungs-Geschwindigkeits-Upgrade können Sie während der Ausführung einer Wiederherstellung aus der S3-Speicherklasse Glacier Flexible Retrieval die Geschwindigkeit der Wiederherstellung erhöhen. Weitere Informationen finden Sie unter [Wiederherstellen archivierter Objekte](#).

26. November 2018

[Wiederherstellen von Ereignisbenachrichtigungen](#)

Amazon-S3-Ereignisbenachrichtigungen unterstützen jetzt Initiierungs- und Abschlussereignisse, wenn Objekte aus der S3-Speicherklasse Glacier Flexible Retrieval wiederhergestellt werden. Weitere Informationen finden Sie unter [Ereignis-Benachrichtigungen](#).

26. November 2018

[PUT direkt zur Speicherklasse S3 Glacier Flexible Retrieval](#)

26. November 2018

Die PUT-Operation von Amazon S3 unterstützt jetzt beim Erstellen von Objekten die Angabe von S3 Glacier Flexible Retrieval als Speicherklasse. Zuvor mussten Sie Objekte aus anderen Amazon-S3-Speicher klassen in die S3-Speich erklasse Glacier Flexible Retrieval übertragen. Wenn Sie S3 Cross-Region Replicati on (CRR) verwenden, können Sie jetzt auch S3 Glacier Flexible Retrieval als Speicherklasse für replizier te Objekte angeben. Weitere Informationen zur Speicherk lasse S3 Glacier Flexible Retrieval finden Sie unter [Speicherklassen](#). Weitere Informationen zum Angeben der Speicherklasse für replizier te Objekte finden Sie unter [Übersicht über die Replikati ons-Konfiguration](#). Weitere Informationen zu den REST API Änderungen, die direkt PUT an S3 Glacier Flexible Retrieval vorgenommen wurden, finden Sie unter [Dokumentenverlauf: PUT direkt zu S3 Glacier Flexible Retrieval](#).

[Neue Speicherklasse](#)

Amazon S3 stellt jetzt eine neue Speicherklasse mit dem Namen S3 Intelligent-Tierung (INTELLIGENT_TIERING) bereit, die für Langzeitanlagen mit veränderlichen oder unbekanntem Zugriffsmustern entwickelt wurde. Weitere Informationen finden Sie unter [Speicherklassen](#).

26. November 2018

[Amazon S3 Block Public Access](#)

Amazon S3 bietet jetzt die Möglichkeit, den öffentlichen Zugriff auf Buckets und Objekte auf Bucket- oder Kontobasis zu blockieren. Weitere Informationen finden Sie unter [Verwenden von Amazon S3 Block Public Access](#).

15. November 2018

[Verbesserungen bei der Filterung in Regeln für die regionsübergreifende Replikation \(\) CRR](#)

In einer CRR Regelkonfiguration können Sie einen Objektfilter angeben, um eine Teilmenge von Objekten auszuwählen, auf die die Regel angewendet werden soll. Zuvor konnten Sie ausschließlich nach Objektschlüsselpräfixen filtern. In dieser Version können Sie nach Objektschlüsselpräfixen, einem oder mehreren Objekt-Markierungen oder beidem filtern. Weitere Informationen finden Sie unter [CRRSetup: Überblick über die Replikationskonfiguration](#).

19. September 2018

[Neue Amazon S3 Select-Funktionen](#)

Amazon S3 Select unterstützt jetzt Apache Parquet-Eingaben, Abfragen zu verschachtelten JSON Objekten und zwei neue CloudWatch Amazon-Überwachungsmetriken (SelectScannedBytes und SelectReturnedBytes).

5. September 2018

[Updates sind jetzt verfügbar über RSS](#)

Sie können jetzt einen RSS Feed abonnieren, um Benachrichtigungen über Aktualisierungen des Amazon S3 S3-Benutzerhandbuchs zu erhalten.

19. Juni 2018

Frühere Updates

In der folgenden Tabelle sind die wichtigen Änderungen in jeder Version des Amazon-S3-Benutzerhandbuchs vor dem 19. Juni 2018 beschrieben.

Änderungen	Beschreibung	Datum
Aktualisierung von Codebeispielen	<p>Aktualisierte Codebeispiele:</p> <ul style="list-style-type: none"> • C# – Alle Beispiele wurden für die Verwendung des aufgabenbasierten asynchronen Musters aktualisiert. Weitere Informationen finden Sie unter Amazon Web Services Asynchronous APIs für .NET in der AWS SDK for .NET Leitfaden für Entwickler. Codebeispiele entsprechen jetzt Version 3 von AWS SDK for .NET. • Java – Alle Beispiele wurden für die Verwendung des Client Builder-Modells aktualisiert. Weitere Informationen zum Client Builder-Modell finden Sie unter Erstellen von Service-Clients. • PHP— Alle Beispiele wurden aktualisiert, sodass sie den verwenden AWS SDK for PHP 3.0. Für weitere Informationen über die AWS SDK for PHP 3.0 finden Sie unter AWS SDK for PHP. • Ruby — Der Beispielcode wurde aktualisiert, sodass die Beispiele mit dem AWS SDK for Ruby Version 3. 	30. April 2018
Amazon S3 meldet jetzt die flexiblen Abruf- und ONEZONE_IA Speicherklassen von S3 Glacier an die Speichermetriken von	<p>Diese Speichermetriken melden nicht nur tatsächliche Byte-Zahlen, sondern beinhalten auch Overhead-Bytes pro Objekt für entsprechende Speicherklassen (ONEZONE_IA , STANDARD_IA und S3 Glacier Flexible Retrieval):</p> <ul style="list-style-type: none"> • Für Objekte der Speicherklassen ONEZONE_IA und STANDARD_IA meldet Amazon S3 Objekte kleiner als 	30. April 2018

Änderungen	Beschreibung	Datum
Amazon CloudWatch Logs	<p>128 KB als 128 KB. Weitere Informationen finden Sie unter Verwenden von Amazon-S3-Speicherklassen.</p> <ul style="list-style-type: none">Für Objekte der Speicherklasse S3 Glacier Flexible Retrieval melden die Speichermetriken die folgenden Overheads:Ein Overhead von 32 KB pro Objekt zu dem Preis, der für Speicherklasse S3 Glacier Flexible Retrieval berechnet wirdEin Overhead von 8 KB pro Objekt zu dem Preis, der für Speicherklasse STANDARD berechnet wird <p>Weitere Informationen finden Sie unter Übergang von Objekten mit Amazon-S3-Lebenszyklus.</p> <p>Weitere Informationen zu Speichermetriken finden Sie unter Metriken mit Amazon überwachen CloudWatch.</p>	
Neue Speicherklasse	<p>Amazon S3 bietet jetzt eine neue Speicherklasse, STANDARD_IA (IA für „Infrequent Access“ (seltener Zugriff)), um Objekte zu speichern. Diese Speicherklasse ist für langlebige Daten und Daten mit weniger häufigem Zugriff optimiert. Weitere Informationen finden Sie unter Verwenden von Amazon-S3-Speicherklassen.</p>	4. April 2018
Amazon S3 Select	<p>Amazon S3 unterstützt jetzt das Abrufen von Objektinh alten auf der Grundlage eines SQL Ausdrucks. Weitere Informationen finden Sie unter Filtern und Abrufen von Daten mit Amazon S3 Select.</p>	4. April 2018

Änderungen	Beschreibung	Datum
Region Asien-Pazifik (Osaka – regional)	<p>Amazon S3 ist jetzt in der Region Asien-Pazifik (Osaka – regional) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>⚠ Important</p> <p>Sie können die Region Asien-Pazifik (Osaka – regional) nur in Verbindung mit der Region Asien-Pazifik (Tokio) verwenden. Um Zugang zur Region Asien-Pazifik (Osaka – regional) anzufordern, wenden Sie sich an Ihren Vertriebsmitarbeiter</p> </div>	12. Februar 2018
Erstellungszeitstempel für Amazon S3 Inventory	Amazon S3 Inventory beinhaltet jetzt einen Zeitstempel mit dem Datum und der Uhrzeit der Erstellung des Amazon-S3-Bestandsberichts. Mithilfe des Zeitstempels können Sie Änderungen in Ihrem Amazon-S3-Speicher ab dem Startzeitpunkt, zu dem der Bestandsbericht erstellt wurde, ermitteln.	16. Januar 2018
Region Europa (Paris)	Amazon S3 ist jetzt in der Region Europa (Paris) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	18. Dezember 2017
Region „China (Ningxia)“	Amazon S3 ist jetzt in der Region China (Ningxia) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	29. November 2017

Änderungen	Beschreibung	Datum
Support für Amazon S3 ORC S3-Lagerbestandsdateien im Format	Amazon S3 unterstützt jetzt das für Apache optimierte Format row columnar (ORC) zusätzlich zum Dateiformat mit kommagetrennten Werten (CSV) für Inventarabgabedateien. Außerdem können Sie jetzt Amazon S3 S3-Inventar standardmäßig abfragen, SQL indem Sie Amazon Athena, Amazon Redshift Spectrum und andere Tools wie Presto, Apache Hive und Apache Spark verwenden. Weitere Informationen finden Sie unter Amazon S3 Inventory .	17. November 2017
Standard-Verschlüsselung für S3-Buckets	Die Amazon-S3-Standard-Verschlüsselung bietet eine Methode zum Festlegen des Verhaltens der Standard-Verschlüsselung für einen S3-Bucket. Sie können die Standard-Verschlüsselung in einem Bucket festlegen, sodass alle Objekte beim Speichern im Bucket verschlüsselt werden. Die Objekte werden mit serverseitiger Verschlüsselung entweder mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) oder AWS verwaltete Schlüssel (SSE-KMS). Weitere Informationen finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets .	06. November 2017
Verschlüsselungsstatus im Amazon S3 Inventory	Amazon S3 unterstützt jetzt das Einsetzen eines Verschlüsselungsstatus in Amazon S3 Inventory, sodass Sie sehen können, wie Ihre Objekte im Ruhezustand für die Überwachung der Compliance von Vorschriften oder andere Zwecke verschlüsselt werden. Sie können Amazon S3 S3-Inventar auch so konfigurieren, dass es mit serverseitiger Verschlüsselung (SSE) oder SSE — verschlüsselt wird, KMS sodass alle Lagerbestandsdateien entsprechend verschlüsselt werden. Weitere Informationen finden Sie unter Amazon S3 Inventory .	06. November 2017

Änderungen	Beschreibung	Datum
Verbesserungen bei der regionsübergreifenden Replikation () CRR	<p>Die regionsübergreifende Replikation unterstützt jetzt Folgendes:</p> <ul style="list-style-type: none">• In einem kontoübergreifenden Szenario können Sie eine CRR Konfiguration hinzufügen, um den Besitz des Replikats zu ändern AWS-Konto dem der Ziel-Bucket gehört. Weitere Informationen finden Sie unter Ändern des Replikat-Eigentümers.• Standardmäßig repliziert Amazon S3 keine Objekte in Ihrem Quell-Bucket, die mithilfe serverseitiger Verschlüsselung mit Schlüsseln erstellt wurden, die in gespeichert sind AWS KMS In Ihrer CRR Konfiguration können Sie Amazon S3 jetzt anweisen, diese Objekte zu replizieren. Weitere Informationen finden Sie unter Replikation verschlüsselter Objekte (SSE-C, SSE -S3, -, SSE -KMS) DSSE KMS.	06. November 2017
Region Europa (London)	Amazon S3 ist jetzt in der Region Europa (London) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	13. Dezember 2016
Region Kanada (Zentral)	Amazon S3 jetzt in der Region Kanada (Zentral) verfügbar Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	8. Dezember 2016

Änderungen	Beschreibung	Datum
Markieren von Objekten	<p>Amazon S3 unterstützt jetzt das Markieren von Objekten. Das Markieren von Objekten ermöglicht Ihnen, Speicher zu kategorisieren. Präfixe für Objektschlüsselnamen ermöglichen Ihnen außerdem, die Speicherung zu kategorisieren. Die Markierung von Objekten fügt ihr eine weitere Dimension hinzu.</p> <p>Die Markierung bietet weitere Vorteile. Dazu zählen:</p> <ul style="list-style-type: none">• Objekt-Tags ermöglichen eine detaillierte Zugriffskontrolle von Berechtigungen (Sie könnten einem IAM Benutzer beispielsweise Berechtigungen für schreibgeschützte Objekte mit bestimmten Tags gewähren).• Differenzierte Kontrolle bei der Spezifizierung einer Lebenszyklus-Konfiguration. Sie können Markierungen spezifizieren, um eine Untermenge von Objekten auszuwählen, auf welche die Lebenszyklus-Regel zutrifft.• Wenn Sie Cross-Region Replication (CRR) konfiguriert haben, kann Amazon S3 die Tags replizieren. Sie müssen der IAM Rolle, die für Amazon S3 erstellt wurde, die erforderlichen Berechtigungen erteilen, damit sie annehmen kann, Objekte in Ihrem Namen zu replizieren.• Sie können CloudWatch Metriken und CloudTrail Ereignisse auch so anpassen, dass Informationen anhand bestimmter Tag-Filter angezeigt werden. <p>Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.</p>	29. November 2016

Änderungen	Beschreibung	Datum
Amazon S3 Lifecycle unterstützt jetzt tagbasierte Filter	Amazon S3 unterstützt jetzt auf Tags basierende Filter bei der Konfiguration des Lebenszyklus. Sie können jetzt Lebenszyklusregeln angeben, in denen Sie ein Schlüsselpräfix, ein oder mehrere Objekt-Markierungen oder eine Kombination aus beidem angeben, um eine Untermenge der Objekte auszuwählen, auf welche die Lebenszyklusregel zutrifft. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten .	29. November 2016
CloudWatch Metriken für Buckets anfordern	Amazon S3 unterstützt jetzt CloudWatch Metriken für Anfragen an Buckets. Wenn Sie diese Metriken für einen Bucket aktivieren, erstellen diese Metriken Berichte in 1-Minuten-Intervallen. Sie können auch konfigurieren, welche Objekte in einem Bucket diese Anforderungsmetriken melden. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch .	29. November 2016
Amazon S3 Inventory	Amazon S3 unterstützt jetzt den Speicherbestand. Amazon S3 Inventory stellt eine Flatfile-Dateiausgabe mit durch Kommata getrennten Werten (.csv) Ihrer Objekte und der zugehörigen Metadaten auf täglicher oder wöchentlicher Basis für einen S3-Bucket oder ein gemeinsames Präfix (d. h. Objekte, deren Namen mit derselben Zeichenfolge beginnen) bereit. Weitere Informationen finden Sie unter Amazon S3 Inventory .	29. November 2016

Änderungen	Beschreibung	Datum
Amazon S3 Analytics – Speicherklassen-Analyse	Die neue analytische Funktion von Amazon S3, die Speicherklassen-Analyse, beobachtet Datenzugriffsmuster, anhand derer Sie entscheiden können, wann Sie STANDARD-Speicher mit weniger häufigem Zugriff in die Speicherklasse STANDARD_IA (IA für „infrequent access“ (seltener Zugriff)) überführen sollen. Wenn die Speicherklassen-Analyse die seltenen Zugriffsmuster für eine gefilterte Datenmenge im Laufe der Zeit beobachtet, können Sie Ihre Lebenszykluskonfigurationen unter Verwendung der Analyseergebnisse verbessern. Diese Funktion beinhaltet auch eine detaillierte tägliche Analyse Ihrer Speichernutzung auf der angegebenen Bucket-, Präfix- oder Tag-Ebene, die Sie in einen S3-Bucket exportieren können.	29. November 2016
Neuer Expedited- und Bulk-Datenabrufe bei der Wiederherstellung archivierter Objekte aus S3 Glacier	Amazon S3 unterstützt jetzt den Expedited- und Bulk-Datenabruf zusätzlich zum Standardabruf bei der Wiederherstellung archivierter Objekte in S3 Glacier. Weitere Informationen finden Sie unter Wiederherstellen eines archivierten Objekts .	21. November 2016
CloudTrail Protokollierung von Objekten	CloudTrail unterstützt die Protokollierung von Amazon S3 API S3-Vorgängen auf Objektebene wie GetObject, PutObject, und DeleteObject. Sie können Ihre Event-Selektoren so konfigurieren, dass sie API Operationen auf Objektebene protokollieren. Weitere Informationen finden Sie unter Protokollieren Amazon S3 API S3-Anrufer mit AWS CloudTrail .	21. November 2016
Region USA Ost (Ohio)	Amazon S3 ist jetzt in der Region USA Ost (Ohio) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	17. Oktober 2016

Änderungen	Beschreibung	Datum
IPv6Unterstützung für Amazon S3 Transfer Acceleration	Amazon S3 unterstützt jetzt Internet Protocol Version 6 (IPv6) für Amazon S3 Transfer Acceleration. Sie können eine Verbindung zu Amazon S3 herstellen, IPv6 indem Sie den neuen Dual-Stack-Endpunkt für Transfer Acceleration verwenden. Weitere Informationen finden Sie unter Erste Schritte mit Amazon S3 Transfer Acceleration .	6. Oktober 2016
IPv6Unterstützung	Amazon S3 unterstützt jetzt Internet Protocol Version 6 (IPv6). Sie können über IPv6 Dual-Stack-Endpunkte auf Amazon S3 zugreifen. Weitere Informationen finden Sie unter Anfragen an Amazon S3 stellen über IPv6 .	11. August 2016
Region Asien-Pazifik (Mumbai)	Amazon S3 ist jetzt in der Region Asien-Pazifik (Mumbai) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	27. Juni 2016
Amazon S3 Transfer Acceleration	Amazon S3 Transfer Acceleration ermöglicht die schnelle, einfache und sichere Übertragung von Dateien über große Entfernungen zwischen Ihrem Client und einem S3-Bucket . Transfer Acceleration nutzt die Vorteile der CloudFront weltweit verteilten Edge-Standorte von Amazon. Weitere Informationen finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration .	19. April 2016
Lebenszyklusunterstützung, um abgelaufene Löschmarkierungen für Objekte zu entfernen	Die Expiration -Aktion der Lebenszykluskonfiguration erlaubt jetzt, Amazon S3 anzuweisen, abgelaufene Löschmarkierungen für Objekte in einem versionsgesteuerten Bucket zu entfernen. Weitere Informationen finden Sie unter Elemente, die Lebenszyklus-Aktionen beschreiben .	16. März 2016

Änderungen	Beschreibung	Datum
Die Bucket-Lebenszyklus-Konfiguration unterstützt jetzt eine Aktion, um unvollständige mehrteilige Uploads abzuberechnen	<p>Die Bucket-Lebenszyklus-Konfiguration unterstützt jetzt die <code>AbortIncompleteMultipartUpload</code> -Aktion. Mit dieser Aktion können Sie Amazon S3 zum Abbrechen mehrteiliger Uploads anweisen, die nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wenn ein mehrteiliger Upload für eine Abbruchoperation in Frage kommt, löscht Amazon S3 alle hochgeladenen Teile und bricht den mehrteiligen Upload ab.</p> <p>Konzeptuelle Informationen finden Sie in folgenden Themen im Amazon-S3-Benutzerhandbuch:</p> <ul style="list-style-type: none">• Abbrechen eines mehrteiligen Uploads• Elemente, die Lebenszyklus-Aktionen beschreiben <p>Die folgenden API Operationen wurden aktualisiert, um die neue Aktion zu unterstützen:</p> <ul style="list-style-type: none">• PUTBucket-Lebenszyklus — Mit der XML Konfiguration können Sie jetzt die <code>AbortIncompleteMultipartUpload</code> Aktion in einer Lebenszyklus-Konfigurationsregel angeben.• Teile auflisten und mehrteiligen Upload initiieren — Beide API Operationen geben jetzt zwei zusätzliche Antwortheader (<code>undx-amz-abort-rule-id</code>) <code>zurückx-amz-abort-date</code> , wenn der Bucket über eine Lebenszyklusregel verfügt, die die <code>AbortIncompleteMultipartUpload</code> Aktion spezifiziert. Diese Header in der Antwort geben an, wann der initiierte mehrteilige Upload für eine Abbruchoperation in Frage kommt, und welche Lebenszyklusregel gilt.	16. März 2016

Änderungen	Beschreibung	Datum
Region Asien-Pazifik (Seoul)	Amazon S3 ist jetzt in der Region Asien-Pazifik (Seoul) verfügbar. Weitere Informationen zu Amazon S3 S3-Regionen und -Endpunkten finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	6. Januar 2016
Neuer Bedingungs-schlüssel und eine Änderung des mehrteiligen Uploads	<p>IAM-Richtlinien unterstützen jetzt einen Amazon S3 <code>s3:x-amz-storage-class</code> S3-Bedingungsschlüssel. Weitere Informationen finden Sie unter Beispiele für Bucket-Richtlinien mit Bedingungs-schlüsseln.</p> <p>Sie müssen nicht mehr der Initiator eines mehrteiligen Uploads sein, um Teile hochzuladen und den Upload abzuschließen. Weitere Informationen finden Sie unter Mehrteiliger Upload API und Berechtigungen.</p>	14. Dezember 2015
Umbenennung der US-Standardregion	Die Regionsnamen-Zeichenfolge hat sich von "USA Standard" in "USA Ost (Nord-Virginia)" geändert. Dies ist nur eine Aktualisierung des Regionsnamens, die Funktionalität bleibt unverändert.	11. Dezember 2015

Änderungen	Beschreibung	Datum
Neue Speicherklasse	<p>Amazon S3 bietet jetzt eine neue Speicherklasse, STANDARD_IA (IA, für seltenen Zugriff), zum Speichern von Objekten. Diese Speicherklasse ist für langlebige Daten und Daten mit weniger häufigem Zugriff optimiert . Weitere Informationen finden Sie unter Verwenden von Amazon-S3-Speicherklassen.</p> <p>Mit Updates für die Lebenszyklus-Konfiguration können Sie jetzt Objekte in die Speicherklasse STANDARD_IA überführen. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.</p> <p>Zuvor hat die Funktion der regionsübergreifenden Replikation die Speicherklasse des Quellobjekts für Objektreplicate verwendet. Wenn Sie jetzt eine regionsübergreifende Replikation konfigurieren, können Sie eine Speicherklasse für das im Ziel-Bucket erstellte Objektreplicate spezifizieren. Weitere Informationen finden Sie unter Überblick über das Replizieren von Objekten.</p>	16. September 2015
AWS CloudTrail Integration	<p>Neu AWS CloudTrail Die Integration ermöglicht es Ihnen, Amazon S3 API S3-Aktivitäten in Ihrem S3-Bucket aufzuzeichnen. Sie können CloudTrail damit die Erstellung oder Löschung von S3-Buckets, Änderungen der Zugriffskontrolle oder Änderungen der Lebenszykluskonfiguration verfolgen. Weitere Informationen finden Sie unter Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail.</p>	1. September 2015

Änderungen	Beschreibung	Datum
Erhöhung des Bucket-Limits	Amazon S3 unterstützt jetzt die Erhöhung des Bucket-Limits. Standardmäßig können Kunden bis zu 100 Buckets in ihren AWS-Konto. Kunden, die zusätzliche Buckets benötigen, können dieses Limit erhöhen, indem sie eine Erhöhung des Service-Limits einreichen. Informationen darüber, wie Sie Ihr Bucket-Limit erhöhen können, finden Sie unter AWS-Service Kontingente in der AWS Allgemein e Referenz. Weitere Informationen erhalten Sie unter Verwendung der AWS SDKs und Bucket-Kontingente, Einschränkungen und Beschränkungen .	4. August 2015
Aktualisierung des Datenkonsistenzmodells	Amazon S3 unterstützt jetzt read-after-write Konsistenz für neue Objekte, die zu Amazon S3 in der Region USA Ost (Nord-Virginia) hinzugefügt wurden. Vor diesem Update unterstützten alle Regionen außer der Region USA Ost (Nord-Virginia) read-after-write Konsistenz für neue Objekte, die auf Amazon S3 hochgeladen wurden. Mit dieser Erweiterung unterstützt Amazon S3 jetzt read-after-write Konsistenz in allen Regionen für neue Objekte, die zu Amazon S3 hinzugefügt wurden. Die read-after-write Konsistenz ermöglicht es Ihnen, Objekte unmittelbar nach der Erstellung in Amazon S3 abzurufen. Weitere Informationen finden Sie unter Regionen .	4. August 2015
Ereignis-Benachrichtigungen	Amazon-S3-Ereignisbenachrichtigungen wurden aktualisiert, um Benachrichtigungen hinzuzufügen, wenn Objekte gelöscht wurden, und das Filtern für Objektnamen mit Präfix- und Suffix-Vergleich hinzuzufügen. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen .	28. Juli 2015

Änderungen	Beschreibung	Datum
CloudWatch Amazon-Integration	Die neue CloudWatch Amazon-Integration ermöglicht es Ihnen, Ihre Amazon S3-Nutzung anhand von CloudWatch Kennzahlen für Amazon S3 zu überwachen und Alarme einzustellen. Unterstützte Metriken sind unter anderem die Gesamtzahl der Bytes für den Standard-Speicher, die Gesamtzahl der Bytes für den Speicher mit reduzierter Redundanz sowie die Gesamtzahl der Objekte für einen bestimmten S3-Bucket. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch .	28. Juli 2015
Unterstützung für das Löschen und Leeren nicht leerer Buckets	Amazon S3 unterstützt jetzt das Löschen und Leeren nicht leerer Buckets. Weitere Informationen finden Sie unter Leeren eines Buckets .	16. Juli 2015
Bucket-Richtlinien für VPC Amazon-Endgeräte	Amazon S3 hat Unterstützung für Bucket-Richtlinien für Virtual Private Cloud (VPC) (VPC) -Endpunkte hinzugefügt. Sie können S3-Bucket-Richtlinien verwenden, um den Zugriff auf Buckets von bestimmten oder bestimmten VPC Endpunkten aus zu steuern. VPCs VPC-Endgeräte sind einfach zu konfigurieren, äußerst zuverlässig und bieten eine sichere Verbindung zu Amazon S3, ohne dass ein Gateway oder eine NAT Instance erforderlich ist. Weitere Informationen finden Sie unter Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien .	29. April 2015
Ereignis-Benachrichtigungen	Amazon S3 S3-Ereignisbenachrichtigungen wurden aktualisiert, um die Umstellung auf ressourcenbasierte Berechtigungen für zu unterstützen AWS Lambda Funktionen. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen .	9. April 2015

Änderungen	Beschreibung	Datum
Regionsübergreifende Replikation	<p>Neu: Amazon S3 unterstützt jetzt regionsübergreifende Replikation. Regionsübergreifende Replikation ist das automatische, asynchrone Kopieren von Objekten über Buckets in verschiedenen Buckets AWS-Regionen. Weitere Informationen finden Sie unter Überblick über das Replizieren von Objekten.</p>	24. März 2015
Ereignis-Benachrichtigungen	<p>Amazon S3 unterstützt jetzt neue Ereignistypen und Ziele in einer Bucket-Benachrichtigungskonfiguration. Vor dieser Version unterstützte Amazon S3 nur den ReducedRedundancyLostObject Ereignistyp s3: und ein SNS Amazon-Thema als Ziel. Weitere Informationen zu den neuen Ereignistypen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.</p>	13. November 2014
Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln	<p>Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS)</p> <p>Amazon S3 unterstützt jetzt serverseitige Verschlüsselung mit AWS KMS. Mit dieser Funktion können Sie den Envelope-Schlüssel verwalten AWS KMS, und Amazon S3 S3-Aufrufe AWS KMS um innerhalb der von Ihnen festgelegten Berechtigungen auf den Envelope-Schlüssel zuzugreifen.</p> <p>Weitere Informationen zur serverseitigen Verschlüsselung mit AWS KMS, siehe Schutz von Daten mithilfe serverseitiger Verschlüsselung mit AWS Key Management Service.</p>	12. November 2014
Region Europa (Frankfurt)	<p>Amazon S3 ist jetzt in der Region Europa (Frankfurt) verfügbar.</p>	23. Oktober 2014

Änderungen	Beschreibung	Datum
Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln	<p>Amazon S3 unterstützt jetzt serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C). Die serverseitige Verschlüsselung ermöglicht Ihnen, Amazon S3 aufzufordern, Ihre ruhenden Daten zu verschlüsseln. Wenn Sie SSE -C verwenden, verschlüsselt Amazon S3 Ihre Objekte mit den von Ihnen bereitgestellten benutzerdefinierten Verschlüsselungsschlüsseln. Amazon S3 übernimmt die Verschlüsselung für Sie, deshalb können Sie von der Verwendung Ihrer eigenen Verschlüsselungsschlüssel profitieren, ohne die Kosten für das Schreiben oder Ausführen Ihres eigenen Verschlüsselungscodes tragen zu müssen.</p> <p>Weitere Informationen zu SSE -C finden Sie unter Serverseitige Verschlüsselung (Verwendung von vom Kunden bereitgestellten Verschlüsselungsschlüsseln).</p>	12. Juni 2014
Lebenszyklus-Unterstützung für das Versioning	<p>Vor diesem Release wurde die Lebenszyklus-Konfiguration nur für nicht versionsfähige Buckets unterstützt. Jetzt können Sie den Lebenszyklus sowohl für nicht versioning-fähige, als auch für versioning-fähige Buckets konfigurieren. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.</p>	20. Mai 2014
Überarbeitung der Themen zur Zugriffskontrolle	<p>Überarbeitung der Dokumentation zur Amazon-S3-Zugriffskontrolle. Weitere Informationen finden Sie unter Identity and Access Management für Amazon S3.</p>	15. April 2014
Überarbeitung des Themas zur Protokollierung des Server-Zugriffs	<p>Überarbeitung der Dokumentation zur Protokollierung des Server-Zugriffs. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.</p>	26. November 2013
. NETSDKBeispiele wurden auf Version 2.0 aktualisiert	<p>. NETSDKDie Beispiele in diesem Handbuch entsprechen jetzt Version 2.0.</p>	26. November 2013

Änderungen	Beschreibung	Datum
SOAPSupport Over HTTP veraltet	SOAPSupport Over HTTP ist veraltet, aber es ist immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Wir empfehlen Ihnen, entweder den REST API oder den zu verwenden AWS SDKs.	20. September 2013
Support für IAM-Richtlinienvariablen	<p>IAMDie Richtliniensprache unterstützt jetzt Variablen . Wenn eine Richtlinie ausgewertet wird, werden alle Richtlinienvariablen durch Werte ersetzt, die von Kontextbasierten Informationen aus der Sitzung des authentifizierten Benutzers bereitgestellt werden. Sie können Richtlinienvariablen zum Definieren von allgemeinen Richtlinien nutzen, ohne explizit alle Komponenten der Richtlinien aufzulisten. Weitere Informationen zu Richtlinienvariablen finden Sie unter Übersicht über IAM Richtlinienvariablen im IAMBenutzerhandbuch.</p> <p>Beispiele für Richtlinienvariablen in Amazon S3 finden Sie unter Beispiele für identitätsbasierte Richtlinien für Amazon S3.</p>	3. April 2013
Konsolensupport für die Zahlung durch den Anforderer	Sie können jetzt Ihren Bucket für Requester Pays unter Verwendung der Amazon-S3-Konsole konfigurieren. Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragung und Nutzung .	31. Dezember 2012

Änderungen	Beschreibung	Datum
Root-Domänen-Unterstützung für das Website-Hosting	<p>Amazon S3 unterstützt jetzt das Hosting statischer Websites in der Root-Domäne. Besucher Ihrer Website können jetzt von ihrem Browser aus auf Ihre Site zugreifen , ohne www in der Webadresse angeben zu müssen (sie können z. B. example.com statt www.example.com verwenden). Viele Kunden hosten bereits statische Websites auf Amazon S3, die von einer www-Subdomäne aus zugänglich sind (z. B. www.example.com). Zuvor mussten Sie für eine Unterstützung des Root-Domänen-Zugriffs Ihren eigenen Webserver betreiben, um die Root-Domänen-Anfragen über einen Proxy von den Browsern auf Ihre Website auf Amazon S3 weiterzugeben. Der Betrieb eines Web-Servers, um Anforderungen über einen Proxy weiterzugeben, führt zusätzliche Kosten, Betriebsaufwand und einen weiteren potenziellen Ausfallpunkt ein. Jetzt können Sie den Vorteil der hohen Verfügbarkeit und Robustheit von Amazon S3 für www- und Root-Domänen-Adressen nutzen. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.</p>	27. Dezember 2012
Überarbeitung der Konsole	Die Amazon-S3-Konsole wurde aktualisiert. Die Dokumentationsthemen, die auf die Konsole verweisen, wurden entsprechend überarbeitet.	14. Dezember 2012

Änderungen	Beschreibung	Datum
Unterstützung für die Archivierung von Daten in S3 Glacier	<p>Amazon S3 unterstützt jetzt eine Speicheroption, die Ihnen gestattet, den kostengünstigen Speicherservice von S3 Glacier für die Datenarchivierung zu nutzen. Für die Archivierung von Objekten definieren Sie Archivierungsregeln, die Objekte identifizieren, sowie einen Zeitrahmen, der festlegt, wann Amazon S3 diese Objekte in S3 Glacier archivieren soll. Sie können die Regeln für einen Bucket ganz einfach mit der Amazon S3-Konsole oder programmgesteuert mit Amazon S3 oder API AWS SDKs.</p> <p>Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.</p>	13. November 2012
Unterstützung für Seitenumleitungen von Websites	<p>Für einen Bucket, der als Website konfiguriert ist, unterstützt Amazon S3 jetzt die Umleitung einer Anfrage für ein Objekt an ein anderes Objekt im selben Bucket oder an ein externes Objekt. URL Weitere Informationen finden Sie unter (Optional) Konfigurieren einer Webseitenumleitung.</p> <p>Weitere Informationen zum Hosting von Websites finden Sie unter Hosten einer statischen Website mit Amazon S3.</p>	4. Oktober 2012
Support für Cross-Origin Resource Sharing (CORS)	<p>Amazon S3 unterstützt jetzt Cross-Origin Resource Sharing (CORS). CORS definiert eine Art und Weise, wie Client-Webanwendungen, die in einer Domain geladen sind, mit Ressourcen in einer anderen Domain interagieren oder auf diese zugreifen können. Mit der CORS Unterstützung in Amazon S3 können Sie umfangreiche clientseitige Webanwendungen auf Amazon S3 aufbauen und selektiv domänenübergreifenden Zugriff auf Ihre Amazon S3 S3-Ressourcen gewähren. Weitere Informationen finden Sie unter Verwendung von ursprungsübergreifendem Ressourcenaustausch (CORS).</p>	31. August 2012

Änderungen	Beschreibung	Datum
Unterstützung von Kostenzuordnungs-Markierungen	Amazon S3 unterstützt jetzt Kostenzuordnungs-Markierungen, um S3-Buckets zu markieren, sodass Sie die Kosten für Projekte oder andere Kriterien einfacher nachverfolgen können. Weitere Informationen zur Verwendung der Markierung von Buckets finden Sie unter Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets .	21. August 2012
Support für MFA - geschützten API Zugriff in Bucket-Richtlinien	<p>Amazon S3 unterstützt jetzt MFA -geschützten API Zugriff, eine Funktion, die erzwingen kann AWS Multi-Faktor-Authentifizierung für ein zusätzliches Maß an Sicherheit beim Zugriff auf Ihre Amazon S3 S3-Ressourcen. Es handelt sich um eine Sicherheitsfunktion, bei der Benutzer den physischen Besitz eines MFA Geräts nachweisen müssen, indem sie einen gültigen MFA Code angeben. Weitere Informationen finden Sie unter AWS Multi-Faktor-Authentifizierung. Sie können jetzt für alle Anfragen zum Zugriff auf Ihre Amazon S3 S3-Ressourcen eine MFA Authentifizierung verlangen.</p> <p>Um die MFA Authentifizierung zu erzwingen, unterstützt Amazon S3 jetzt die <code>aws:MultiFactorAuthAge</code> Key-in-a-Bucket-Richtlinie. Ein Beispiel für eine Bucket-Richtlinie finden Sie in Beispiel Erforderlich MFA.</p>	10. Juli 2012
Unterstützung des Ablaufens von Objekten	Sie können das Ablaufen von Objekten nutzen, um das automatische Entfernen von Daten nach einem konfigurierten Zeitraum einzuplanen. Sie legen die Ablaufzeit eines Objekts fest, indem Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen.	27. Dezember 2011
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region Südamerika (São Paulo). Weitere Informationen finden Sie unter Zugreifen auf einen Amazon-S3-Bucket und Auflisten des Buckets .	14. Dezember 2011

Änderungen	Beschreibung	Datum
Multi-Object Delete	Amazon S3 unterstützt jetzt Multi-Object DeleteAPI, sodass Sie mehrere Objekte in einer einzigen Anfrage löschen können. Mit dieser Funktion können Sie eine große Anzahl von Objekten schneller aus Amazon S3 entfernen, als wenn Sie mehrere einzelne DELETE Anfragen verwenden würden. Weitere Informationen finden Sie unter Löschen von Amazon-S3-Objekten .	7. Dezember 2011
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region USA West (Oregon). Weitere Informationen finden Sie unter Buckets und Regionen .	8. November 2011
Aktualisierung der Dokumentation	Korrektur von Fehlern in der Dokumentation	8. November 2011
Aktualisierung der Dokumentation	Neben den Korrekturen von Fehlern in der Dokumentation enthält diese Version auch die folgenden Erweiterungen: <ul style="list-style-type: none">• Neue serverseitige Verschlüsselungsabschnitte mit dem AWS SDK for PHP und die AWS SDK for Ruby (siehe Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) angeben).	17. Oktober 2011

Änderungen	Beschreibung	Datum
Unterstützung der serverseitigen Verschlüsselung	<p>Amazon S3 unterstützt jetzt die serverseitige Verschlüsselung. Diese ermöglicht Ihnen, Amazon S3 aufzufordern, Ihre ruhenden Daten zu verschlüsseln, d. h. Ihre Objektdaten zu verschlüsseln, wenn Amazon S3 Ihre Daten in seinen Rechenzentren auf Datenträger schreibt. Zusätzlich zu den REST API Updates AWS SDK for Java und .NET stellen die erforderlichen Funktionen bereit, um serverseitige Verschlüsselung anzufordern. Sie können beim Hochladen von Objekten auch serverseitige Verschlüsselung anfordern, indem Sie den AWS Management Console. Weitere Informationen zur Datenverschlüsselung finden Sie unter Datenverschlüsselung verwenden.</p>	4. Oktober 2011
Aktualisierung der Dokumentation	<p>Neben den Korrekturen von Fehlern in der Dokumentation enthält diese Version auch die folgenden Erweiterungen:</p> <ul style="list-style-type: none">• Ruby und PHP Beispiele wurden dem Senden von Anforderungen Abschnitt hinzugefügt.• Es wurden Abschnitte hinzugefügt, in denen beschrieben wird, wie Presigned URLs generiert und verwendet wird. Weitere Informationen erhalten Sie unter Objekte mit vorsignierten teilen URLs und Objekte mit vorsignierten teilen URLs.• Ein vorhandener Abschnitt wurde aktualisiert, um Folgendes einzuführen: AWS Explorer für Eclipse und Visual Studio. Weitere Informationen finden Sie unter Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs.	22. September 2011

Änderungen	Beschreibung	Datum
Unterstützung von Anforderungen unter Verwendung temporärer Sicherheitsanmeldeinformationen	<p>Zusätzlich zur Verwendung Ihres AWS-Konto und IAM Benutzer-Sicherheitsanmeldedaten, um authentifizierte Anfragen an Amazon S3 zu senden, können Sie jetzt Anfragen mit temporären Sicherheitsanmeldedaten senden, die Sie erhalten von AWS Identity and Access Management (IAM). Sie können das AWS Security Token Service API oder der AWS SDK Wrapper-Bibliotheken, von IAM denen Sie diese temporären Anmeldeinformationen anfordern können. Sie können diese temporären Anmeldeinformationen für Ihre eigene Verwendung erhalten, oder sie verbundenen Benutzern und Anwendungen erteilen. Diese Funktion ermöglicht es Ihnen, Ihre Benutzer außerhalb zu verwalten AWS und geben Sie ihnen temporäre Sicherheitsanmeldeinformationen für den Zugriff auf Ihre AWS Ressourcen schätzen.</p> <p>Weitere Informationen finden Sie unter Senden von Anforderungen.</p> <p>Weitere Informationen zur IAM Unterstützung temporärer Sicherheitsanmeldedaten finden Sie unter Temporäre Sicherheitsanmeldedaten im IAM Benutzerhandbuch.</p>	3. August 2011

Änderungen	Beschreibung	Datum
<p>Multipart Upload API wurde erweitert und ermöglicht nun das Kopieren von Objekten mit bis zu 5 TB</p>	<p>Vor dieser Version API unterstützte Amazon S3 das Kopieren von Objekten mit einer Größe von bis zu 5 GB. Um das Kopieren von Objekten zu ermöglichen, die größer als 5 GB sind, erweitert Amazon S3 jetzt den API mehrteiligen Upload um einen neuen Vorgang. <code>UploadPart (Copy)</code> Sie können diese Operation für einen mehrteiligen Upload verwenden, um Objekte mit einer Größe von 5 TB zu kopieren. Weitere Informationen finden Sie unter Objekte kopieren, verschieben und umbenennen.</p> <p>Konzeptionelle Informationen zum mehrteiligen Upload finden Sie API unter. Hochladen und Kopieren von Objekten mit mehrteiligen Uploads</p>	<p>21. Juni 2011</p>
<p>SOAPAPIAnrufe über HTTP sind deaktiviert</p>	<p>Um die Sicherheit zu erhöhen, SOAP API HTTP sind Call-Over deaktiviert. Authentifizierte und anonyme SOAP Anfragen müssen mit SSL an Amazon S3 gesendet werden.</p>	<p>6. Juni 2011</p>
<p>IAM ermöglicht eine kontoübergreifende Delegierung</p>	<p>Bisher benötigte ein IAM Benutzer für den Zugriff auf eine Amazon S3 S3-Ressource die Berechtigungen der beiden übergeordneten Ressourcen AWS-Konto und der Besitzer der Amazon S3 S3-Ressource. Beim kontoübergreifenden Zugriff benötigt der IAM Benutzer jetzt nur noch die Erlaubnis des Eigentümerkontos. Das heißt, wenn ein Ressourcenbesitzer Zugriff auf ein gewährt AWS-Konto, der AWS-Konto kann seinen IAM Benutzern nun Zugriff auf diese Ressourcen gewähren.</p> <p>Weitere Informationen finden Sie im Benutzerhandbuch unter Erstellen einer Rolle zur Delegierung von Berechtigungen an einen IAM IAM Benutzer.</p> <p>Weitere Informationen über die Angabe von Prinzipalen in einer Bucket-Richtlinie finden Sie unter Prinzipien für Bucket-Richtlinien.</p>	<p>6. Juni 2011</p>

Änderungen	Beschreibung	Datum
Neuer Link	Die Endpunktinformationen dieses Dienstes befinden sich jetzt im AWS Allgemeine Referenz. Weitere Informationen finden Sie unter Regionen und Endpunkte in der AWS Allgemeine Referenz .	1. März 2011
Unterstützung des Hostings statischer Websites in Amazon S3	Amazon S3 führt die erweiterte Unterstützung des Hostings statischer Websites ein. Dies beinhaltet die Unterstützung von Indextdokumenten und eines benutzerdefinierten Fehlerdokuments. Wenn Sie diese Funktionen verwenden , geben Anforderungen an das Stammverzeichnis Ihres Buckets oder eines Unterordners (z. B. <code>http://mywebsite.com/subfolder</code>) Ihr Indextdokument statt der Liste der Objekte in Ihrem Bucket zurück. Wenn ein Fehler festgestellt wird, gibt Amazon S3 Ihre benutzerdefinierte Fehlermeldung statt einer Amazon-S3-Fehlermeldung zurück. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3 .	6. Juni 2011
Die Endpunktinformationen dieses Dienstes befinden sich jetzt im AWS Allgemeine Referenz. Weitere Informationen finden Sie unter Regionen und Endpunkte in der AWS Allgemeine Referenz .	1. März 2011	

Änderungen	Beschreibung	Datum
Unterstützung des Hostings statischer Websites in Amazon S3	Amazon S3 führt die erweiterte Unterstützung des Hostings statischer Websites ein. Dies beinhaltet die Unterstützung von Indext Dokumenten und eines benutzerdefinierten Fehlerdokuments. Wenn Sie diese Funktionen verwenden , geben Anforderungen an das Stammverzeichnis Ihres Buckets oder eines Unterordners (z. B. <code>http://mywebsite.com/subfolder</code>) Ihr Indext Dokument statt der Liste der Objekte in Ihrem Bucket zurück. Wenn ein Fehler festgestellt wird, gibt Amazon S3 Ihre benutzerdefinierte Fehlermeldung statt einer Amazon-S3-Fehlermeldung zurück. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3 .	17. Februar 2011
APISupport für Response-Header	Das GET Objekt ermöglicht es Ihnen REST API jetzt, die Antwortheader der REST GET Objektanforderung für jede Anfrage zu ändern. Das bedeutet, Sie können Objekt-Metadaten in der Antwort ändern, ohne das eigentliche Objekt zu ändern. Weitere Informationen finden Sie unter Herunterladen von Objekten .	14. Januar 2011
Unterstützung für große Objekte	In Amazon S3 wurde die maximale Größe für ein Objekt, das Sie in einem S3-Bucket speichern können, von 5 GB auf 5 TB erhöht. Wenn Sie das verwenden RESTAPI, können Sie Objekte mit bis zu 5 GB in einem einzigen PUT Vorgang hochladen. Bei größeren Objekten müssen Sie den Multipart Upload verwenden RESTAPI, um Objekte in Teilen hochzuladen. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads .	9. Dezember 2010
Mehrteiliger Upload	Der mehrteilige Upload unterstützt schnellere, flexiblere Uploads in Amazon S3. Mithilfe dieser Funktion können Sie ein einzelnes Objekt in mehreren Teilen hochladen . Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads .	10. November 2010

Änderungen	Beschreibung	Datum
Unterstützung kanonischer IDs in Bucket-Richtlinien	Sie können jetzt kanonische Richtlinien IDs in Bucket-Richtlinien angeben. Weitere Informationen finden Sie unter Prinzipien für Bucket-Richtlinien	17. September 2010
Amazon S3 funktioniert mit IAM	Dieser Service ist jetzt integriert mit AWS Identity and Access Management (IAM). Weitere Informationen finden Sie unter AWS-Services mit denen Sie IAM im IAMBenutzerhandbuch arbeiten.	2. September 2010
Benachrichtigungen	Mit der Amazon S3-Benachrichtigungsfunktion können Sie einen Bucket so konfigurieren, dass Amazon S3 eine Nachricht zu einem Amazon Simple Notification Service (AmazonSNS) -Thema veröffentlicht, wenn Amazon S3 ein wichtiges Ereignis in einem Bucket erkennt. Weitere Informationen finden Sie unter Einrichten von Benachrichtigungen für Bucket-Ereignisse .	14. Juli 2010
Bucket-Richtlinien	Bucket-Richtlinien sind ein Zugriffsmanagementsystem, mit dem Sie Zugriffsberechtigungen für Buckets, Objekte und Objektmengen festlegen. Diese Funktionalität ergänzt Zugriffsrichtlinien und ersetzt sie in vielen Fällen. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3 .	6. Juli 2010
Path-Style-Syntax in allen Regionen verfügbar	Amazon S3 unterstützt jetzt die Pfad-Syntax für alle Buckets in der Region US Classic, oder wenn sich der Bucket innerhalb derselben Region befindet wie der Endpunkt der Anfrage. Weitere Informationen finden Sie unter Virtuelles Hosting .	9. Juni 2010
Neuer Endpunkt für Europa (Irland)	Amazon S3 bietet jetzt einen Endpunkt für Europa (Irland): <code>http://s3-eu-west-1.amazonaws.com</code>	9. Juni 2010

Änderungen	Beschreibung	Datum
Konsole	Sie können Amazon S3 jetzt verwenden über AWS Management Console. Informationen zu allen Funktionen von Amazon S3 in der Konsole finden Sie im Amazon Simple Storage Service-Benutzerhandbuch.	9. Juni 2010
Reduzierte Redundanz	Amazon S3 ermöglicht Ihnen jetzt, Ihre Speicherkosten zu reduzieren, indem Objekte in Amazon S3 mit reduzierter Redundanz gespeichert werden. Weitere Informationen finden Sie unter Reduced Redundancy Storage .	12. Mai 2010
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region Asien-Pazifik (Singapur). Weitere Informationen finden Sie unter Buckets und Regionen .	28. April 2010
Objekt-Versioning	In dieser Version wird ein Objekt-Versioning eingeführt. Alle Objekte können jetzt einen Schlüssel und eine Version haben. Wenn Sie das Versioning für einen Bucket aktivieren, gibt Amazon S3 allen Objekten, die einem Bucket hinzugefügt werden, eine eindeutige Versions-ID. Diese Funktion ermöglicht Ihnen eine Wiederherstellung nach einem unbeabsichtigtem Überschreiben oder Löschen. Weitere Informationen finden Sie unter Versioning und Verwenden von Versioning .	8. Februar 2010
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region USA West (Nordkalifornien). Der neue Endpunkt für Anfragen an diese Region ist <code>s3-us-west-1.amazonaws.com</code> . Weitere Informationen finden Sie unter Buckets und Regionen .	2. Dezember 2009

Änderungen	Beschreibung	Datum
AWS SDK for .NET	AWS bietet jetzt Bibliotheken, Beispielcode, Tutorials und andere Ressourcen für Softwareentwickler, die es vorziehen, Anwendungen mithilfe von zu erstellen . NETsprachspezifische API Operationen anstelle von oder. REST SOAP Diese Bibliotheken bieten grundlegende Funktionen (die nicht im REST Oder enthalten sind SOAPAPIs) wie Anforderungsauthentifizierung, erneute Anforderungsversuche und Fehlerbehandlung, sodass der Einstieg erleichtert wird. Weitere Informationen zu sprachspezifischen Bibliotheken und Ressourcen finden Sie unter Entwickeln mit Amazon S3 unter Verwendung der AWS SDKs .	11. November 2009

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.